



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR – TE 145561

***SISTEM HUMAN MACHINE INTERFACE PADA UNIVERSAL
TESTING MACHINE***

Juniar Adi Prasetya
NRP 10311500000013

Dosen Pembimbing
Slamet Budiprayitno, ST., MT

DEPARTEMEN TEKNIK ELEKTRO OTOMASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT – TE 145561

***HUMAN MACHINE INTERFACE SYSTEM FOR UNIVERSAL
TESTING MACHINE***

Juniar Adi Prasetya
NRP 10311500000013

Advisor
Slamet Budiprayitno, ST., MT

***DEPARTMENT OF ELECTRICAL AUTOMATION ENGINEERING
Faculty of Vocations
Institut Teknologi Sepuluh Nopember
Surabaya 2018***

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**SISTEM HUMAN MACHINE INTERFACE PADA UNIVERSAL TESTING MACHINE**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 26 Juli 2018



Jumar Adi Prasetya
NRP 10311500000013

-----Halaman ini sengaja dikosongkan-----

**SISTEM HUMAN MACHINE INTERFACE PADA UNIVERSAL
TESTING MACHINE**

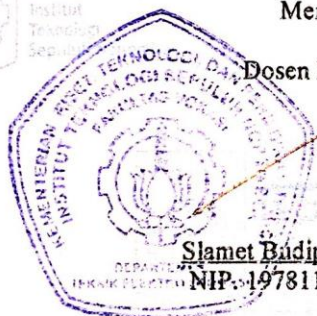
TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Memperoleh Gelar Ahli Madya Teknik
Pada**

**Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember**

Menyetujui :

Dosen Pembimbing



Slamet Badipravitno, ST., MT.
NIP. 19781111 320102 1 002

**SURABAYA
JULI, 2018**

-----Halaman ini sengaja dikosongkan-----

SISTEM HUMAN MACHINE INTERFACE PADA UNIVERSAL TESTING MACHINE

Nama : Juniar Adi Prasetya
Pembimbing : Slamet Budiprayitno, ST., MT.

ABSTRAK

Universal Testing Machine (UTM) merupakan sebuah alat untuk menguji material logam atau material lainnya agar tercapainya standarisasi yang diinginkan. Dalam pengujian material tersebut dapat menggunakan uji tarik dan uji tekan. Namun untuk dapat menggambarkan proses uji secara keseluruhan, diperlukan suatu sistem monitoring yang konstan. Karena pada proses monitoring merupakan salah satu acuan untuk melakukan proses selanjutnya terhadap suatu material. Pada penelitian ini sistem *Human Machine Interface* (HMI) diterapkan pada sebuah mesin UTM untuk memantau hasil pembacaan berupa angka perubahan posisi dan berat yang kemudian dirubah menjadi grafik secara *real time*.

Untuk itu penulis merancang sistem HMI pada mesin UTM dengan menggunakan *software Visual Basic* untuk merepresentasikan grafik proses keseluruhan pada pengujian tekan, sehingga kesalahan dalam pengambilan data saat proses pengujian berlangsung dapat berkurang. Pengguna juga dapat menyimpan data hasil pengujian tersebut pada *MS Excel*.

Dari pengujian yang sudah dilakukan, sistem HMI pada mesin UTM yang sudah dibuat dapat memudahkan pengguna dalam melakukan monitoring saat melakukan pengujian tekan. Saat proses pengujian berlangsung data dan grafik dapat terbaca secara *real time*. *Moving average* filter yang sudah diterapkan pada sistem HMI pada mesin UTM juga dapat meredam *noise* yang ada.

Kata Kunci : *Human Machine Interface, Universal Testing Machine, Visual Basic, Mikrokontroler*

-----Halaman ini sengaja dikosongkan-----

HUMAN MACHINE INTERFACE SYSTEM FOR UNIVERSAL TESTING MACHINE

Name : Juniar Adi Prasetya
Advisor : Slamet Budiprayitno, ST., MT.

ABSTRACT

Universal Testing Machine (UTM) is a tool for testing metal materials or other materials to achieve the desired standardization. In testing the material can use tensile test and press test. In testing the material can use tensile test and compression test. But to be able to describe the overall test process, a constant monitoring system is required. Because the monitoring process is one of the reference to do the next process of a material. In this study HMI (Human Machine Interface) system is applied to a UTM machine to monitor the stress and strain which then changed into graph in real time.

For that the author will use HMI system on UTM machine by using Visual Basic software to represent every graphical process at compression test, so that errors in data retrieval during the testing process can be reduced. Users can store the test result data in MS Excel.

From the test that has been done, HMI systems on UTM machines that have been created to facilitate users in monitoring when performing compression tests. During the testing process the data and graphics can be read in real time. Moving average filters that have been applied to the HMI system on a UTM engine can dampen the noise.

Keywords : Human Machine Interface, Universal Testing Machine, Visual Basic, Microcontroller

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Segala puji dan syukur kehadiran Allah SWT atas segala rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul: ***“SISTEM HUMAN MACHINE INTERFACE PADA UNIVERSAL TESTING MACHINE”***

Tugas Akhir ini disusun sebagai salah satu persyaratan untuk memperoleh gelar ahli madya pada jurusan Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember. Pada kesempatan ini penulis ingin mengucapkan terima kasih kepada berbagai pihak yang telah memberikan bantuan dan dukungan dalam penyelesaian Tugas Akhir, terutama kepada :

1. Bapak, Ibu, dan keluarga yang senantiasa memberikan semangat dan doa yang tidak terbatas.
2. Bapak Slamet Budiprayitno, ST., MT. selaku dosen pembimbing yang telah banyak memberikan bimbingan, saran, serta masukan yang sangat berarti bagi penulis.
3. Seluruh dosen Departemen Teknik Elektro Otomasi yang telah banyak memberikan ilmu selama penulis menempuh kuliah.
4. Teman-teman di Departemen Teknik Elektro ITS yang tidak dapat penulis sebutkan satu persatu, terutama rekan-rekan DE-10 yang selalu memberikan sumbangan pemikiran serta semangat dalam menyelesaikan Tugas Akhir ini.
5. Teman-teman DEPARTEMEN HMKI yang selalu memberikan semangat dan motivasi tersendiri bagi penulis.
6. Dan semua pihak yang tidak bisa penulis sebutkan.

Besar harapan penulis bahwa buku Tugas Akhir ini dapat memberikan informasi dan manfaat bagi pembaca pada umumnya dan mahasiswa Departemen Teknik Elektro pada khususnya.

Surabaya, 26 Juli 2018

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL	i
HALAMAN JUDUL	iii
PERNYATAAN KEASLIAN TUGAS AKHIR	v
HALAMAN PENGESAHAN	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
 BAB I PENDAHULUAN	 1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Metodologi Penelitian	2
1.6 Sistematika Laporan	3
 BAB II TEORI DASAR	 5
2.1 <i>Universal Testing Machine</i> [2]	5
2.2 <i>Human Machine Interface</i> [3]	6
2.3 Komunikasi Serial [4]	6
2.4 <i>Visual Basic .NET</i> [5]	8
2.5 Filter Digital <i>Moving Average</i> [6]	11
2.6 <i>Analog to Digital Converter</i> [7]	12
2.7 STM32F103C [8]	13
 BAB III PERANCANGAN DAN PEMBUATAN ALAT	 17
3.1 Blok Fungsional Sistem	17
3.2 Perancangan Perangkat Lunak	18
3.2.1 Perancangan Program <i>Software HMI</i>	18
3.2.1.1 <i>Flowchart Form Login</i>	20
3.2.1.2 <i>Flowchart Form Sign Up</i>	20
3.2.1.3 <i>Flowchart Form Main Menu</i>	21
3.2.1.4 <i>Flowchart Form Information</i>	22
3.2.1.5 <i>Flowchart Form Result</i>	23

3.2.1.6 Tampilan HMI Pada Aplikasi UTM	24
3.2.2 Perancangan Program Mikrokontroler	32
BAB IV PENGUJIAN DAN ANALISA DATA	37
4.1 Pengujian Penerimaan Data Tekanan Dari Sensor <i>Load Cell</i> ...	37
4.2 Pengujian Penerimaan Data Perubahan Posisi	40
4.3 Pengujian Filter Digital <i>Moving Average</i>	42
BAB V PENUTUP	45
5.1 Kesimpulan	45
5.2 Saran	45
DAFTAR PUSTAKA	47
LAMPIRAN A LISTING PROGRAM	49
A. <i>Listing</i> Program Mikrokontroler	49
B. <i>Listing</i> Program <i>Visual Basic Form Login</i>	51
C. <i>Listing</i> Program <i>Visual Basic Form Sign Up</i>	52
D. <i>Listing</i> Program <i>Visual Basic Form Main Menu</i>	56
E. <i>Listing</i> Program <i>Visual Basic Form Information</i>	66
F. <i>Listing</i> Program <i>Visual Basic Form Result</i>	67
LAMPIRAN B DATASHEET	71
A. <i>Datasheet</i> Mikrokontroller STM32F103C8T6	71
DAFTAR RIWAYAT HIDUP	73

DAFTAR GAMBAR

	HALAMAN
Gambar 2. 1 Bentuk Fisik Universal Testing Machine	5
Gambar 2. 2 Sinyal Transmisi Sinkron	7
Gambar 2. 3 Penggunaan Option Explicit Pada Program.....	9
Gambar 2. 4 Penggunaan Imports Statements Pada Program.....	9
Gambar 2. 5 Macam-macam Cara Penulisan Main	10
Gambar 2. 6 Contoh Penggunaan Main Procedure	10
Gambar 2. 7 Hasil Eksekusi Dari Penggunaan Main Procedure.....	10
Gambar 2. 8 ADC Dengan Kecepatan Sampling Rendah dan Kecepatan Sampling Tinggi.....	12
Gambar 2. 9 STM32F103 Atau Blue Pill	13
Gambar 2. 10 Memasukkan URL untuk flash STM32F103.....	14
Gambar 2. 11 Menginstal board STM32F103 ke Arduino IDE.....	15
Gambar 2. 12 Pin Koneksi ST-Link V2 Dengan Board STM32F103	15
Gambar 3. 1 Diagram Blok Fungsional Sistem Keseluruhan.....	17
Gambar 3. 2 Data Flow Diagram HMI	19
Gambar 3. 3 <i>Flowchart Form Login</i>	20
Gambar 3. 4 <i>Flowchart Form Sign Up</i>	21
Gambar 3. 5 <i>Flowchart Form Main Menu</i>	22
Gambar 3. 6 <i>Flowchart Form Information</i>	23
Gambar 3. 7 <i>Flowchart Form Result</i>	23
Gambar 3. 8 Tampilan Project Baru	24
Gambar 3. 9 Tampilan <i>Form Login</i>	25
Gambar 3. 10 Tampilan <i>Form Sign Up</i>	26
Gambar 3. 11 Tampilan <i>Form Main Menu</i>	28
Gambar 3. 12 Tampilan <i>Form Information</i>	30
Gambar 3. 13 Tampilan <i>Form Result</i>	31
Gambar 3. 14 Flowchart Perancangan <i>Moving Average Filter</i>	33
Gambar 3. 15 Listing Program Deklarasi Variabel dan Tipe Data ..	33
Gambar 3. 16 Listing Program Inisialisasi pada <i>Void Setup</i>	34
Gambar 3. 17 Listing Program Fungsi <i>Moving Average Filter</i>	35
Gambar 3. 18 Listing Program Pengiriman Data Mikrokontroler dengan Komputer	35
Gambar 4. 1 Tampilan pada HMI Saat Proses Pengujian Load Cell Berlangsung.....	39

Gambar 4. 2 Tampilan pada HMI Saat Proses Pengujian Perubahan
Posisi Berlangsung.....42

Gambar 4. 3 Tampilan Grafik Perbandingan Data dengan Filter
Moving Average dan Data Tanpa Filter.43

DAFTAR TABEL

	HALAMAN
Tabel 3. 1 Komponen Tampilan HMI Pada Form <i>Login</i>	25
Tabel 3. 2 Komponen Tampilan HMI Pada Form Sign Up.....	27
Tabel 3. 3 Komponen Tampilan HMI Pada Form <i>Main Menu</i>	29
Tabel 3. 4 Komponen Tampilan HMI Pada Form Information	30
Tabel 3. 5 Komponen Tampilan HMI Pada Form Result.....	31
Tabel 4. 1 Hasil Pengujian Penerimaan Data Load Cell	38
Tabel 4. 2 Hasil Pengujian Penerimaan Data Perubahan Posisi	40

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Banyak perusahaan yang bergerak di bidang industri logam, karet, besi, dan baja yang membutuhkan pengujian terhadap material guna mengetahui spesifikasi bahan atau material yang akan dipasarkan. Karena sebuah kualitas produk tidak dapat di produksi dengan mengandalkan perkiraan saja. Pengujian yang dilakukan untuk menentukan karakteristik suatu bahan agar tercapainya standarisasi yang diinginkan diantaranya dengan melakukan uji tarik (*tensile test*) dan uji tekan (*compression test*). Salah satu alat yang dapat digunakan untuk melakukan pengujian tersebut yaitu *Universal Testing Machine* (UTM). Tetapi dengan menggunakan alat uji ini saja tidak cukup, tanpa sebuah sistem monitoring yang konstan. Dibutuhkan sebuah sistem *Human Machine Interface* (HMI) pada mesin UTM agar dapat menggambarkan proses uji secara keseluruhan. Data hasil pengujian yang diperoleh dari alat ini akan dikonversikan dari analog ke digital dengan menggunakan mikrokontroler. Data inilah yang kemudian dianalisa lebih lanjut menggunakan komputer untuk mendapatkan parameter-parameter yang diinginkan.

Permasalahan yang dapat dilihat yaitu dengan banyaknya perusahaan yang membutuhkan pengujian terhadap produknya, maka dibutuhkan alat uji dengan sistem monitoring secara *real time* ataupun *hystorical*. Untuk itu penulis akan merancang sebuah sistem HMI pada mesin UTM.

Sistem HMI yang berupa *Graphic User Interface* (GUI) pada suatu tampilan layar komputer akan berhadapan langsung dengan operator mesin atau pengguna untuk memonitoring secara *real time*. Salah satu sistem HMI yang penulis gunakan yaitu dengan *software Visual Basic*. Data yang sudah didapat dari UTM akan dikonversikan dan diintegrasikan dengan komputer menggunakan mikrokontroler. Pada *Visual Basic* akan merepresentasikan grafik proses keseluruhan sehingga dapat mengetahui letak titik luluh dan titik tegangan tarik maksimum pada objek yang telah diuji. Dan pengguna dapat menyimpan data hasil pengujian tersebut pada MS Excel 2016. Dengan adanya sistem HMI pada mesin UTM ini diharapkan pengguna dapat lebih mudah dalam melakukan

monitoring pengujian tarik, sehingga dapat menjadi alternatif sistem monitoring pada mesin UTM.

1.2 Permasalahan

Saat proses pengujian material pada UTM berlangsung dibutuhkan sistem monitoring untuk melihat data hasil pembacaan sensor yang dikeluarkan secara terus menerus, sehingga dapat mengetahui perubahan nilai yang pasti dan juga untuk merepresentasikan data yang terbaca menjadi grafik untuk mengetahui letak titik tegangan maksimum pada objek yang diuji.

1.3 Batasan Masalah

Agar penulisan buku Tugas Akhir ini tidak menyimpang dan mengambang dari tujuan yang semula direncanakan sehingga mempermudah mendapatkan data dan informasi yang diperlukan, maka penulis menetapkan batasan-batasan masalah sebagai berikut:

- a. Variabel proses yang dimonitor adalah perubahan posisi dan berat.
- b. Sistem HMI yang dibuat hanya memonitoring mesin UTM yang sudah ada sebelumnya.
- c. HMI yang dibuat digunakan untuk memonitoring proses uji tekan pada mesin UTM.

1.4 Tujuan

Penelitian ini bertujuan untuk memonitoring proses pengujian beton pada UTM, sehingga dapat mempermudah dalam pengambilan data dan dapat merepresentasikan data grafik yang dihasilkan oleh alat uji untuk ditampilkan pada *PC/Laptop*.

1.5 Metodologi Penelitian

Penelitian ini dilakukan melalui beberapa tahapan metodologi, yaitu, tahap persiapan, tahap perencanaan dan pembuatan alat, tahap pengujian dan analisis, dan yang terakhir adalah penyusunan laporan berupa buku Tugas Akhir.

Pada tahap studi literatur ini akan dilakukan pencarian data, bahan, dan literatur. Dimana literatur diperoleh dari Tugas Akhir dengan Judul Rancang Bangun Sistem *Human Machine Interface* (HMI) Pada Miniplant Pembangkit Listrik Tenaga Mikrohidro Skala Laboratorium, Program Studi D3 Teknik Instrumentasi, Jurusan

Teknik Fisika, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember[1]. Dari beberapa artikel yang didapatkan, memunculkan sebuah ide untuk membuat judul Sistem *Human Machine Interface* Pada *Universal Testing Machine*. Kemudian dipelajari mengenai bagaimana cara kerja *Universal Testing Machine*, studi tentang mikrokontroler STM32F103C dan karakteristik baik secara hardware dan software, mempelajari filter digital, mempelajari parsing data untuk memilah semua data yang sudah didapatkan pada mikrokontroler

Pada tahap perancangan sistem terdapat perancangan *software*. Untuk tahap ini pembuatan *software Human Machine Interface* menggunakan *software Visual Studio 2015*, sehingga data yang sudah dikirim ke PC/Laptop menggunakan kabel USB dari mikrokontroler dapat ditampilkan menjadi grafik, yang kemudian data tersebut juga dapat disimpan pada *MS Excel 2016*. Data percobaan yang telah diperoleh selanjutnya akan dianalisis. Dari hasil analisis, akan ditarik kesimpulan dari penelitian yang telah dilakukan. Tahap akhir penelitian adalah penyusunan laporan penelitian.

1.6 Sistematika Laporan

Sistematika pembahasan Tugas Akhir ini terdiri dari lima bab, yaitu pendahuluan, teori penunjang, perencanaan dan pembuatan alat, pengujian dan analisa alat, serta penutup.

BAB I : PENDAHULUAN

Membahas tentang latar belakang, permasalahan, batasan masalah, maksud dan tujuan, sistematika laporan, serta relevansi.

BAB II : TEORI DASAR

Bab ini menjelaskan teori dasar yang berisi tentang konsep yang dijadikan landasan dan mendukung dalam perencanaan dan pembuatan alat meliputi pengertian *Human Machine Interface*, *Universal Testing Machine*, *Software Visual Studio 2015*, dan mikrokontroler *STM 32F103C*.

BAB III : PERANCANGAN ALAT

Membahas tentang perencanaan dan pembuatan perangkat lunak yang meliputi program yang akan digunakan untuk memonitoring mesin UTM.

BAB IV : PENGUJIAN DAN ANALISA ALAT

Membahas tentang pengukuran, pengujian, dan penganalisaan terhadap kepresisian pembacaan alat ukur yang telah dibuat.

BAB V : PENUTUP

Menjelaskan tentang kesimpulan dari Tugas Akhir ini dan saran-saran untuk pengembangan alat ini lebih lanjut.

BAB II

TEORI DASAR

Bab ini membahas mengenai teori dasar dari peralatan yang digunakan dalam Pembuatan Perangkat Lunak Sistem *Human Machine Interface* Pada *Universal Testing Machine*, antara lain: Pengertian dari *Human Machine Interface*, *Universal Testing Machine*, *Software Visual Studio 2015*, Filter Digital, Struktur Program dan mikrokontroler *STM 32F1*.

2.1 *Universal Testing Machine* [2]

UTM atau *Universal Testing Machine* merupakan mesin uji gaya sumbu tunggal tarik atau tekan yang sering digunakan dalam pengujian material logam maupun material lainnya yang memiliki karakteristik yang dipengaruhi oleh gaya yang diberikan pada material tersebut. *Universal Testing Machine* ini dapat melakukan pengujian bahan atau material seperti karet, beton, besi, logam, dan baja baik itu dengan uji tarik (*Tensile Test*) maupun dengan uji tekan (*Compression Test*). UTM akan memberikan informasi mengenai seberapa besar pengukuran yang akan diuji terhadap bahan, sehingga standarisasi yang diinginkan dapat tercapai dengan sempurna. Data yang langsung diperoleh dari UTM ini adalah perubahan panjang sampel terhadap setiap besar gaya yang diberikan. Hasil ini akan dikonversikan ke dalam bentuk grafik. Data awal inilah yang kemudian dianalisa lebih lanjut menggunakan komputer untuk mendapatkan parameter-parameter yang sebelumnya telah didapatkan. Pada Gambar 2.1 merupakan contoh bentuk fisik mesin UTM di S1 Teknik Sipil ITS.



Gambar 2. 1 Bentuk Fisik Universal Testing Machine

2.2 Human Machine Interface [3]

HMI (*Human Machine Interface*) adalah sistem yang menghubungkan antara manusia dan teknologi mesin. HMI dapat berupa pengendali dan visualisasi status baik dengan manual maupun melalui visualisasi komputer yang bersifat *real time*. Sistem HMI biasanya bekerja secara *online* dan *real time* dengan membaca data yang dikirimkan melalui *I/O port* yang digunakan oleh sistem kontroler-nya. *Port* yang biasanya digunakan untuk kontroler dan akan dibaca oleh HMI antara lain adalah *port com*, *port USB*, *port RS232* dan ada pula yang menggunakan *port serial*. Tugas dari HMI (*Human Machine Interface*) yaitu membuat visualisasi dari teknologi atau sistem secara nyata. Sehingga dengan desain HMI dapat disesuaikan sehingga memudahkan pekerjaan fisik. Tujuan dari HMI adalah untuk meningkatkan interaksi antara mesin dan operator melalui tampilan layar komputer dan memenuhi kebutuhan pengguna terhadap informasi sistem.

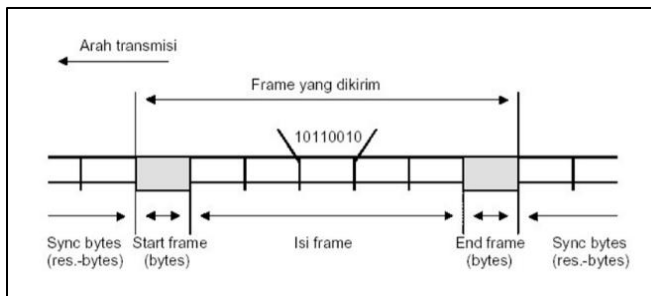
HMI dalam dunia industri berupa suatu tampilan *GUI* (*Graphic User Interface*) pada suatu tampilan layar komputer yang akan dihadapi oleh operator mesin maupun pengguna yang membutuhkan data kerja mesin. HMI terdapat berbagai macam visualisasi untuk monitoring dan data mesin yang terhubung secara *online* dan *real time*. Pada HMI terdapat visualisasi pengendali mesin berupa tombol, slider, dan sebagainya yang dapat difungsikan untuk mengendalikan mesin sebagaimana mestinya. Sebagai tambahan, HMI juga menampilkan data-data rangkuman kerja mesin termasuk secara grafik.

2.3 Komunikasi Serial [4]

Komunikasi serial ialah pengiriman data secara serial (data dikirim satu persatu secara berurutan), sehingga komunikasi serial jauh lebih lambat daripada komunikasi paralel. Komunikasi Serial dapat digunakan untuk menggantikan komunikasi paralel jalur data 8-bit dengan baik. Agar komunikasi serial dapat bekerja dengan baik, data byte harus diubah ke dalam bit-bit serial menggunakan peralatan yang disebut shift register parallel-in serial-out, kemudian data dikirimkan hanya dengan satu jalur data saja. Hal yang serupa dikerjakan pada penerima, dimana penerima harus mengubah bit-bit serial yang diterimanya menjadi data byte yang persis seperti data

semula pada pengirim, dengan menggunakan shift register serial-in parallel-out.

Komunikasi data serial mengenal dua buah metode, yaitu synchronous dan asynchronous. Metode synchronous mengirimkan datanya beberapa byte atau karakter (atau disebut blok data atau frame) sebelum meminta konfirmasi apakah data sudah diterima dengan baik atau tidak. Sementara metode asynchronous data dikirim satu byte setiap pengiriman. Biasanya tidak dibutuhkan konfirmasi penerimaan data. Dari kedua jenis metode tersebut dapat dipilih dan dilakukan lewat program. Namun dewasa ini proses pengiriman data serial tersebut sudah dilakukan oleh sebuah chip tersendiri (*Hardware*). Salah satu chip disebut *UART (Universal Asynchronous Reciever Transmitter)* dan satunya lagi disebut *USART (Universal Synchronous Asynchronous Reciever Transmitter)*. Dalam protokol berbeda, synchronous memerlukan sinyal tambahan yang digunakan untuk men-synchron-isasi setiap denyut dari proses transfer. Gambar 2.2 adalah contoh bentuk pengiriman sinyal transmisi sinkron.



Gambar 2. 2 Sinyal Transmisi Sinkron

Komunikasi data serial Asynchronous sekarang sudah digunakan demikian luas untuk transmisi yang berorientasi karakter, sementara metode Synchronous digunakan untuk transmisi yang berorientasi blok. Pada mode Asynchronous, setiap karakter ditempatkan berada diantara bit start dan bit stop. Bit start selalu satu bit, tapi stop bit bisa satu bit atau dua bit. Start bit selalu 0 (*low*) dan stop bit selalu 1 (*high*).

Pada komunikasi serial Asynchronous, peralatan dan modem dapat diprogram untuk menggunakan lebar data 7 atau 8-bit. Tentu

saja ditambah dengan Stop bit. Dahulu, system karakter ASCII masih terbatas pada data 7-bit, namun sekarang ASCII extended sudah lazim menggunakan lebar data 8-bit. Pada peralatan kuno, dengan komponen-komponen yang lambat pula, dibutuhkan stop bit yang agak panjang, hal ini dimaksudkan untuk memberi kesempatan peralatan untuk menangani data yang telah diterimanya, dan untuk mempersiapkan diri untuk menerima byte berikutnya. Namun sekarang modem PC kita dewasa ini biasanya hanya menggunakan satu bit stop. Jika kita hitung, dengan menggunakan satu bit stop total kita memiliki 10-bit untuk setiap karakter 8-bit. Dengan kata lain setiap karakter 8-bit dikirim bersama-sama start dan stop bit sehingga total menjadi 10-bit, yang artinya ada proses transfer lebih panjang 20% setiap karakternya.

Pada sistem yang sangat mementingkan integritas data yang disimpan, maka ditambahkan lah bit paritas kepada bingkai data tersebut. Maksudnya untuk setiap karakter 8-bit kita masih menambahkan bit paritas disamping bit start dan bit stop. Sehingga total adalah 11-bit. Adapun bit paritas adalah bit yang menunjukkan bahwa data yang dimaksud adalah memiliki jumlah bit 1s (*high*) ganjil atau genap. Bit paritas adalah bit di luar data yang bersangkutan atau merupakan tambahan. Chip *UART* khusus biasanya sudah dilengkapi dengan keperluan paritas tersebut secara hardware. Bahkan ada beberapa pilihan untuk penanganan paritas ini, misalnya *odd*-, *even*- dan *no-parity*.

2.4 Visual Basic .NET [5]

Visual Basic .NET (atau VB.NET) merupakan salah satu bahasa pemrograman yang bisa digunakan untuk membangun aplikasi-aplikasi.NET di platform Microsoft .NET. Tidak seperti generasi sebelumnya Visual Basic versi 6.0 ke bawah yang lebih difokuskan untuk pengembangan aplikasi desktop, Visual Basic .NET memungkinkan para pengembang membangun bermacam aplikasi, baik desktop maupun aplikasi web. Seiring dengan perkembangan aplikasi perangkat lunak yang semakin kompleks.

Sebagai bahasa yang telah dirancang kembali dari awal, suatu program Visual Basic memiliki urutan yang merupakan struktur dari program yang akan dibuat. Urutan struktur program tersebut meliputi Option statements, Imports statements, dan Main procedure.

Option statements merupakan perintah option yang berfungsi untuk menentukan aturan dasar penanganan kode dalam program pada saat proses kompilasi berlangsung. Sebuah variabel yang sudah ada dalam program dapat ditentukan proses apa yang akan terjadi saat program dikompilasi. Salah satu contoh penggunaannya yaitu Option Explicit, menentukan apakah variabel-variabel yang digunakan dalam program harus dideklarasikan terlebih dahulu atau tidak, yang defaultnya adalah On. Berikut merupakan contoh program dari Option Explicit pada Gambar 2.3.

```
Option Explicit Off

Module Modmain
    Sub Main()
        a = 100000
    End Sub
End Module
```

Gambar 2. 3 Penggunaan Option Explicit Pada Program

Yang jika dikompilasi tidak menyebabkan kesalahannya walaupun variabel “a” tidak pernah dideklarasikan. Dan struktur program Visual Basic berikutnya adalah Imports Statements, yaitu perintah imports yang berfungsi untuk memudahkan pengetikan tanpa harus menetik nama class di dalam namespace yang di import. Berikut merupakan contoh dari cara penggunaan Imports Statements.

```
Imports System
Module Hello
    Sub Main()
        Console.WriteLine("Hello World !")
    End Sub
End Module
```

Gambar 2. 4 Penggunaan Imports Statements Pada Program

Pada Gambar 2.4 merupakan contoh penggunaan Imports Statements. Yang setelah adanya Imports System penulisan Console.WriteLine menjadi seperti Gambar 2.4. Dalam hal ini kita tidak perlu lagi menetik System.Console.WriteLine, tapi cukup Console.WriteLine, karena kita telah melakukan Imports System.

Struktur program yang terakhir yaitu Main Procedure. Procedure Main merupakan titik awal dari program anda – merupakan procedure yang pertama kali dieksekusi ketika anda menjalankan kode anda. Jadi apa yang ingin anda jalankan pertama kali pada program anda ditempatkan pada procedure main ini. Ada empat cara penulisan Main, yaitu sebagai Sub atau Function yang memiliki parameter atau tidak.

- Sub Main()
- Sub Main(ByVal CmdArgs() As String)
- Function Main() As Integer
- Function Main(ByVal CmdArgs() As String) As Integer

Gambar 2. 5 Macam-macam Cara Penulisan Main

Bentuk procedure yang paling sering digunakan adalah Sub Main() seperti pada contoh Gambar2.6.

```
Imports System
Module Hello
    Sub Main(ByVal CmdArgs() As String)
        Dim i As Integer
        Console.WriteLine("Jumlah Parameter :" & CmdArgs.Length())
        For i = 0 to CmdArgs.Length()-1
            Console.WriteLine("Parameter :" & i.ToString() & "adalah :"
& CmdArgs(i))
        Next
    End Sub
End Module
```

Gambar 2. 6 Contoh Penggunaan Main Procedure

Pada Gambar 2.6 merupakan contoh penggunaan dari main procedure pada program. Jika program tersebut dieksekusi maka akan menghasilkan sebagai berikut.

```
Coba Hello World
Jumlah Parameter :2
Parameter :0 adalah :Hello
Parameter :1 adalah :World
```

Gambar 2. 7 Hasil Eksekusi Dari Penggunaan Main Procedure

Pada Gambar 2.7 merupakan hasil penggunaan main procedure pada program yang telah dieksekusi. Dalam hal ini kita menjalankan program Coba.exe pada command prompt dengan

mengirim dua parameter yaitu kata Hello dan World yang dianggap sebagai array bertipe string.

2.5 Filter Digital Moving Average [6]

Filter digital adalah suatu prosedur matematika/ algoritma yang mengolah sinyal masukan digital dan menghasilkan isyarat keluaran digital yang memiliki sifat tertentu sesuai dengan tujuan filter. Filter digital dapat dibagi menjadi dua yaitu Filter Digital IIR (infinite impulse response) dan FIR (finite impulse response). Pembagian ini berdasarkan pada tanggapan impuls filter tersebut. FIR memiliki tanggapan impuls yang panjangnya terbatas, sedangkan IIR tidak terbatas. FIR sering juga disebut sebagai filter non rekursif dan IIR sebagai filter rekursif. FIR tidak memiliki pole, maka kestabilan dapat dijamin sedangkan IIR memiliki pole-pole sehingga lebih tidak stabil. Pada filter digital orde tinggi, kesalahan akibat pembulatan koefisien filter dapat mengakibatkan ketidakstabilan.

Sesuai dengan namanya, filter ini mengambil nilai rata-rata beberapa sampel sinyal yang berdekatan dan hasil yang diperoleh kurang lebih adalah sinyal aslinya. Filter ini cocok digunakan jika teknik enkripsi yang digunakan sekedar berupa penambahan *noise* pada sinyal aslinya. Sebab secara kasarnya, hasil dari *Moving Average Filter* tersebut “meredam” *noise* yang ditambahkan tadi sehingga memungkinkan penerima memperoleh informasi aslinya.

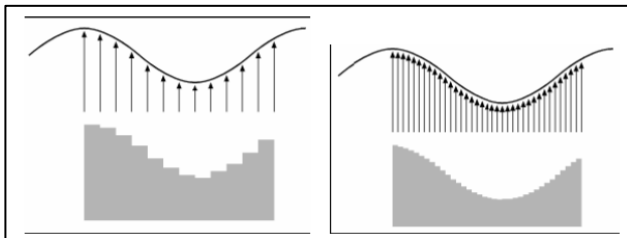
Maksud dari “meredam” tersebut adalah mengurangi *noise* yang ditambahkan tadi sehingga informasi yang dikirimkan dapat diperoleh kembali. Tentu saja sinyal tersebut tidak 100% dapat diperoleh kembali dengan jelas, sebab masih ada unsur *noise* yang terselubung di dalamnya. Selain itu, masih ada sinyal asli yang ikut “diredam” oleh filter ini. Namun, setidaknya untuk alasan keamanan, 60% dari sinyal yang dapat diolah sudah dianggap cukup baik. Persamaan moving average filter dapat dilihat pada Persamaan (2.1) berikut ini.

$$y[n] = \frac{x[n] + x[n - 1] + x[n - 2] + \cdots + x[n - 9]}{10} \dots\dots\dots(2.1)$$

Dari Persamaan (2.1), setiap sampel berikutnya adalah rata-rata dari 10 sampel sebelumnya. Sehingga hasil sampel yang dihitung adalah berupa hasil perkiraan atau prediksi yang diperoleh dengan membandingkan sampel-sampel sebelumnya.

2.6 Analog to Digital Converter [7]

Analog to Digital Converter (ADC) adalah pengubah input analog menjadi kode – kode digital. ADC banyak digunakan sebagai Pengatur proses industri, komunikasi digital dan rangkaian pengukuran/ pengujian. Umumnya ADC digunakan sebagai perantara antara sensor yang kebanyakan analog dengan sistem komputer seperti sensor suhu, cahaya, tekanan/ berat, aliran dan sebagainya kemudian diukur dengan menggunakan sistem digital (komputer). ADC (*Analog to Digital Converter*) memiliki 2 karakter prinsip, yaitu kecepatan sampling dan resolusi. Kecepatan sampling suatu ADC menyatakan seberapa sering sinyal analog dikonversikan ke bentuk sinyal digital pada selang waktu tertentu. Kecepatan sampling biasanya dinyatakan dalam *sample per second* (SPS).



Gambar 2. 8 ADC Dengan Kecepatan Sampling Rendah dan Kecepatan Sampling Tinggi

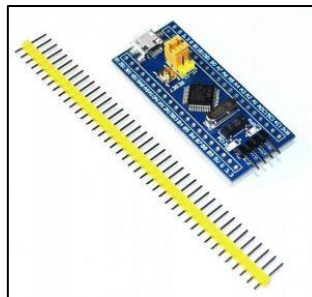
Resolusi ADC menentukan ketelitian nilai hasil konversi ADC. Sebagai contoh: ADC 8 bit akan memiliki output 8 bit data digital, ini berarti sinyal input dapat dinyatakan dalam 255 ($2^n - 1$) nilai diskrit. ADC 12 bit memiliki 12 bit output data digital, ini berarti sinyal input dapat dinyatakan dalam 4096 nilai diskrit. Dari contoh diatas ADC 12 bit akan memberikan ketelitian nilai hasil konversi yang jauh lebih baik daripada ADC 8 bit. Prinsip kerja ADC adalah mengkonversi sinyal analog ke dalam bentuk besaran

yang merupakan rasio perbandingan sinyal input dan tegangan referensi. Sebagai contoh, bila tegangan referensi 5 volt, tegangan input 3 volt, rasio input terhadap referensi adalah 60%. Jadi, jika menggunakan ADC 8 bit dengan skala maksimum 255, akan didapatkan sinyal digital sebesar $60\% \times 255 = 153$ (bentuk decimal) atau 10011001 (bentuk biner).

$$\begin{aligned}
 \text{Signal} &= (\text{sample}/\text{max_value}) * \text{reference_voltage} \\
 &= (153/255) * 5 \\
 &= 3 \text{ Volt}
 \end{aligned}
 \dots\dots\dots(2.2)$$

2.7 STM32F103C [8]

STM32F103C merupakan sebuah papan pengembangan yang memiliki nama lain “Blue Pill”. STM32F103C adalah modul dari ST Electronic berbasiskan mikrokontroler arsitektur ARM 32bit dengan prosesor Cortex-M3. Untuk *Chip* STM32F103C mempunyai banyak fitur yang dapat digunakan, seperti port USB, dua port serial, pin PWM 16 bit, dan pin ADC 12 bit. Mikrokontroler ini bekerja pada tegangan 3,3V. Gambar 2.9 adalah bentuk fisik dari STM32F103C.



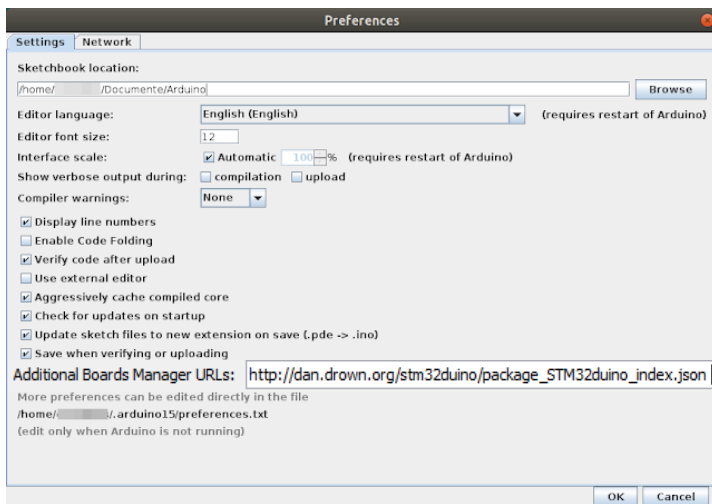
Gambar 2. 9 STM32F103 Atau Blue Pill

Untuk pembuatan program pada board ini dapat dipermudah dengan menggunakan *software* Arduino IDE. Tetapi sebelum memasukkan program pada mikrokontroler, sebuah bootloader Arduino harus di flash kedalam board STM32F103. Proses ini dapat

dilakukan melalui port serial atau menggunakan interface debug pada MCU dengan bantuan ST-Link tool.

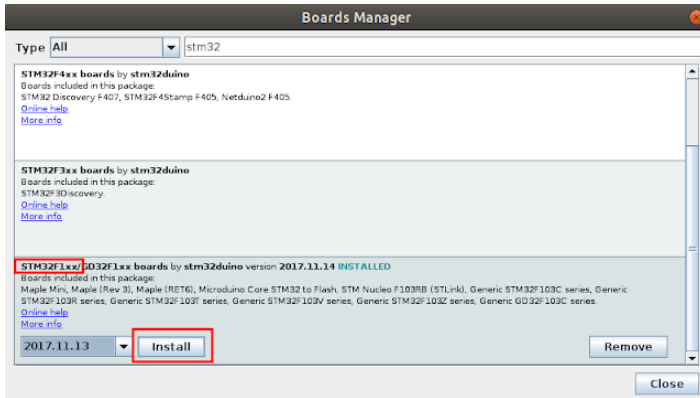
Pada software Arduino IDE masih belum memiliki dukungan default untuk board STM32F103. Maka dari itu, agar board STM32F103 bisa diintegrasikan dengan Arduino IDE harus di setting terlebih dahulu. Berikut adalah langkah-langkah untuk semua sistem operasi penggunaan board STM32F103 dengan Arduino IDE.

Untuk langkah pertama jalankan *software* Arduino IDE, kemudian pilih File – Preferences, dan tambahkan URL berikut ini pada board manager yang tersedia **“http://dan.drown.org/stm32duino/package_STM32duino_index.json”** klik OK untuk menutup dialog. Langkah tersebut dapat dilihat pada Gambar 2.11.



Gambar 2. 10 Memasukkan URL untuk flash STM32F103

Langkah selanjutnya, pilih Tools – Board Manager. Didalam dialog yang muncul, cari STM32 dan pilih STM32F1XX untuk “Blue Pill”. Jika board menggunakan chipset yang berbeda (contoh : STM32F3) pilih seperti yang tertera pada board lalu install. Penjelasan langkah tersebut dapat dilihat pada Gambar 2.11.



Gambar 2. 11 Menginstal board STM32F103 ke Arduino IDE

Sebelum benar – benar mengunggah sketches ke dalam board, dibutuhkan bootloader STM32duino. Disini menggunakan sebuah ST – Link V2 clone untuk memasukkan bootloadernya. Koneksi ke board STM32F103 sangat mudah, menggunakan SWCLK, SWDIO, GND dan pin 3,3V. sambungkan semua menggunakan kabel jumper yang sudah tersedia pada ST – Link clone. Pindahkan jumper untuk set BOOT0 to 1. Berikut merupakan pin koneksi antara ST-Link V2 dengan board STM32F103 dapat dilihat pada Gambar 2.12.

ST-LINK V2		Papan STM32F103C8T6
SWCLK	>>	DCLK
SWDIO	>>	DIO
GND	>>	GND
3.3V	>>	3.3V

Gambar 2. 12 Pin Koneksi ST-Link V2 Dengan Board STM32F103

-----Halaman ini sengaja dikosongkan-----

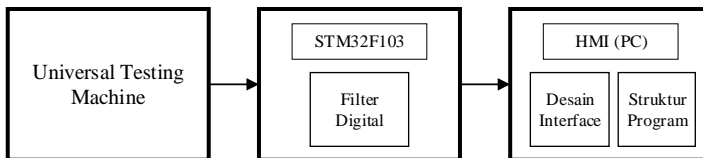
BAB III

PERANCANGAN DAN PEMBUATAN ALAT

Pada bab ini berisi mengenai tahapan yang dilakukan dalam perencanaan dan pembuatan Tugas Akhir dengan judul Sistem *Human Machine Interface* pada *Universal Testing Machine*. Penjelasan diawali dengan blok fungsional sistem secara keseluruhan pada Gambar 3.1 yang meliputi proses kerja alat dalam bentuk alur diagram. Secara keseluruhan sistem kerja alat ini dikontrol menggunakan mikrokontroler STM32F103 yang terhubung dengan alat ukur *Universal Testing Machine* dan menggunakan komunikasi serial *USB*. Adapun penjelasan tentang perancangan *software* yang membahas desain *interface* dan struktur program yang digunakan.

3.1 Blok Fungsional Sistem

Sebelum melakukan perancangan sistem dalam pembuatan Tugas Akhir ini, diperlukan sebuah perancangan blok fungsional sistem berupa blok diagram yang menjelaskan sistem kerja secara keseluruhan dari sistem human machine interface pada universal testing machine. Perancangan sistem bertujuan untuk memberikan gambaran secara umum kepada pemakai tentang sistem HMI yang akan dibuat. Secara keseluruhan blok fungsional sistem dapat dilihat pada Gambar 3.1 berikut.



Gambar 3.1 Diagram Blok Fungsional Sistem Keseluruhan

Sesuai dengan diagram blok pada Gambar 3.1 menjelaskan sistem kerja HMI pada UTM secara keseluruhan. Saat awal mesin dinyalakan, semua data yang keluar dari mesin akan dikirim ke mikrokontroler untuk diolah terlebih dahulu. Proses yang dilakukan mikrokontroler pertama kali yaitu melakukan *Digital Filtering* pada tiap data yang sudah diterima. Data berupa tegangan yang masuk pada mikrokontroler akan difilter menggunakan metode *Moving*

Average Filter untuk menghilangkan *noise* saat pengiriman data berlangsung. Kemudian data yang sudah difilter akan dikonversi agar dapat mudah terbaca oleh pengguna. Setelah seluruh proses pada mikrokontroler selesai, data yang sudah diolah tadi akan langsung dikirimkan ke komputer dengan komunikasi serial kabel *USB*. Pada komputer akan dibuat suatu desain tampilan *Human Machine Interface* untuk UTM menggunakan *software Visual Studio 2015*. Dari desain interface dan struktur program yang sudah dibuat pada komputer, data hasil pengukuran yang terbaca dari mikrokontroler akan ditampilkan pada tampilan HMI kemudian diolah lagi menjadi grafik agar dapat memudahkan pengguna dalam melakukan penelitian terhadap objek yang sudah diuji.

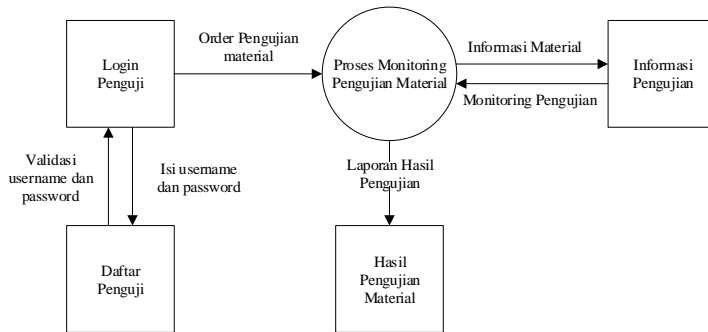
Adapun tampilan (*display*) *interface* yang disajikan diantaranya gambar grafik yang berjalan selama proses pengukuran berlangsung, kenaikan perubahan tegangan dan regangan secara bertahap. Kemudian juga ditampilkan titik luluh, tegangan maksimum, dan regangan maksimum. Setelah sistem telah selesai melakukan pengukuran, dari data yang sudah di dapatkan maka secara otomatis sistem akan melakukan perhitungan tegangan leleh, tegangan ultimat, delta, luas, dan elongasi.

3.2 Perancangan Perangkat Lunak

Dalam perancangan perangkat lunak untuk sistem *Human Machine Interface* pada *Universal Testing Machine*, terdapat beberapa program yang digunakan untuk melakukan monitoring UTM, baik program pada mikrokontroler STM32F103 dan juga program untuk pembuatan tampilan HMI pada komputer menggunakan *software Visual Studio 2015*.

3.2.1 Perancangan Program Software HMI

Perancangan HMI dengan *software Visual Studio 2015* digunakan untuk pembuatan tampilan interface pada komputer, sehingga pengguna dapat memonitor mesin pada saat proses pengujian berlangsung. Pada pembuatan sistem HMI ini dibagi beberapa *form* dengan fungsi khusus masing-masing. *Form* tersebut diantaranya adalah *Form Login*, *Form Daftar*, *Form Main menu*, *Form Informasi*, dan *Form Hasil Pengukuran*. Berikut juga disajikan diagram blok sistem pada Gambar 3.2 yang digunakan dalam tugas akhir ini.

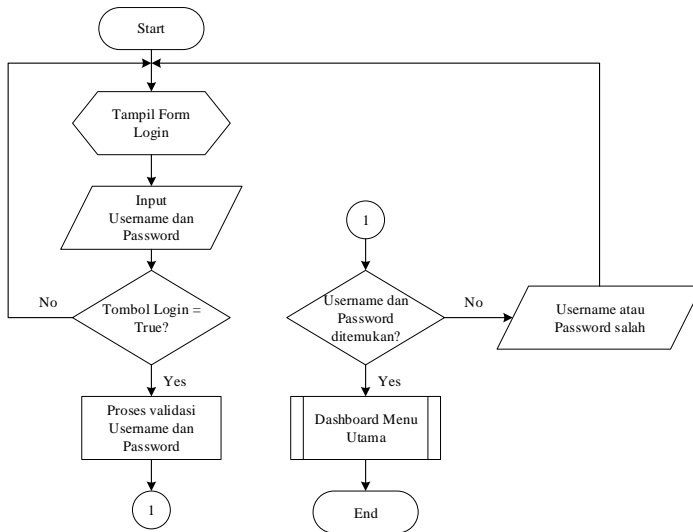


Gambar 3. 2 Data Flow Diagram HMI

Berdasarkan data flow diagram pada Gambar 3.2 menjelaskan bagaimana alur kerja dari HMI yang sudah dibuat. Saat awal program dijalankan maka akan muncul form *login* yang berfungsi agar pengguna dapat mengakses aplikasi setelah dilakukan validasi berupa username dan password. Jika kita belum memiliki akun untuk mengakses aplikasi, maka kita dapat membuat akun baru dengan masuk pada form daftar. Pada form daftar pengguna harus mengisi kan data username dan password. Jika sudah mendapatkan akses form utama akan keluar, semua proses monitoring dilakukan pada form utama, langkah awal koneksikan HMI dengan mikrokontroler yang sudah terpasang dengan memilih port dan baudrate yang tersedia. Sebelum melakukan pengukuran kita harus mengisi form informasi terlebih dahulu, setelah data yang diperlukan terisi semua kembali ke form *main menu* dan pengukuran dapat dilakukan dengan menekan tombol *start*. Setelah proses pengukuran cukup maka tekan tombol *stop*, kemudian untuk melihat laporan hasil pengujian masuk ke dalam form hasil, pada form ini seluruh data hasil pengukuran dan grafik akan ditampilkan.

Adapun penjelasan *flowchart* dari setiap form yang telah dibuat dalam sistem HMI pada UTM. Bagan alir (*flowchart*) adalah bagan (chart) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi (Burch, J.G., 1992). Dan berikut merupakan *flowchart* dan penjelasan dari setiap form dalam sistem HMI pada UTM.

3.2.1.1 Flowchart Form Login



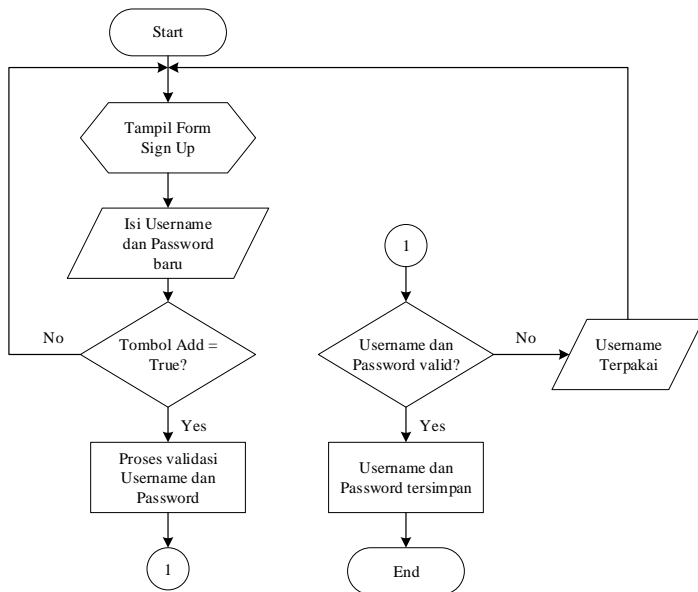
Gambar 3.3 Flowchart Form Login

Gambar 3.3 menjelaskan *flowchart* dari *form login* yang diawali dengan tampilnya *form login*, kemudian pengguna harus mengisi data *login* yaitu berupa *username* dan *password* yang sudah terdaftar. Apabila *username* dan *password* yang dimasukkan sudah sesuai dengan data dalam database, maka proses selanjutnya akan menuju ke *dashboard form main menu*. Apabila *username* dan *password* yang dimasukkan tidak sesuai atau tidak ada dalam database maka akan muncul pemberitahuan bahwa data yang diisikan tidak terdapat pada database dan kembali ke *form login*.

3.2.1.2 Flowchart Form Sign Up

Pada *form* ini berfungsi untuk menambahkan akun jika pengguna belum memiliki akun untuk mengakses aplikasi ini. Dalam *form* ini terdapat database yang berguna untuk menyimpan setiap akun yang sudah didaftarkan. Untuk membuat akun harus mengisi *username* dan *password* terlebih dahulu, jika *username* sudah terpakai maka akan muncul pemberitahuan untuk mengganti

username baru. Gambar 3.4 menjelaskan *flowchart* pada form *Sign UP*.

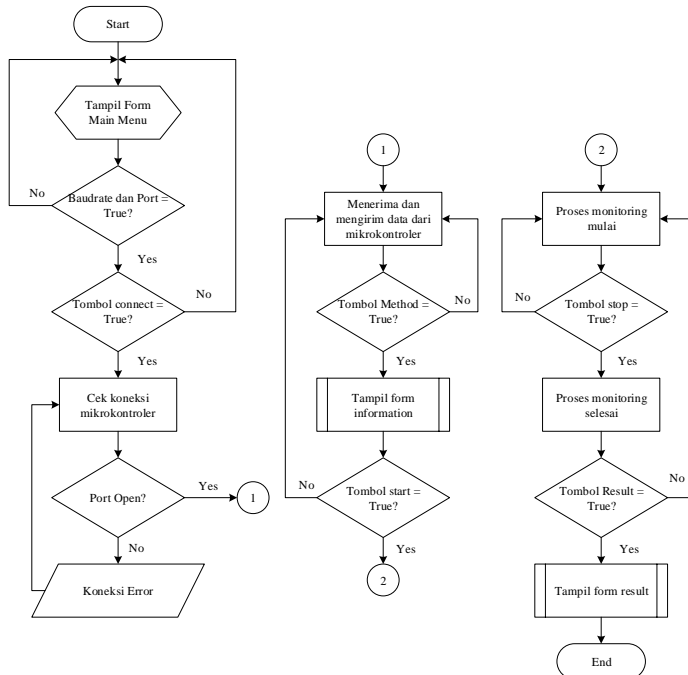


Gambar 3. 4 *Flowchart Form Sign Up*

3.2.1.3 *Flowchart Form Main Menu*

Gambar 3.5 menjelaskan *flowchart* dari form *main menu*, dimana semua proses pengukuran atau monitoring dilakukan pada form ini. Saat awal form muncul, langkah awal yang dilakukan yaitu dengan memilih setting port dan baudrate pada *combobox* untuk mengkoneksikan HMI dengan mikrokontroler dan pilih tombol *connect*. Setelah HMI terkoneksi dengan mikrokontroler, maka pengguna harus mengisi form *information* terlebih dahulu dengan menekan tombol *method*. Jika proses tersebut sudah terpenuhi kembali ke form *main menu*. Proses pengukuran dapat dilakukan dengan menekan tombol *start* dan HMI akan menerima data dari mikrokontroler kemudian mengubah data tersebut menjadi bentuk grafik secara *real time*, jika proses pengukuran selesai tekan tombol *stop* dan HMI akan menghentikan penerimaan data dari mikrokontroler. Untuk melihat laporan hasil monitoring akhir,

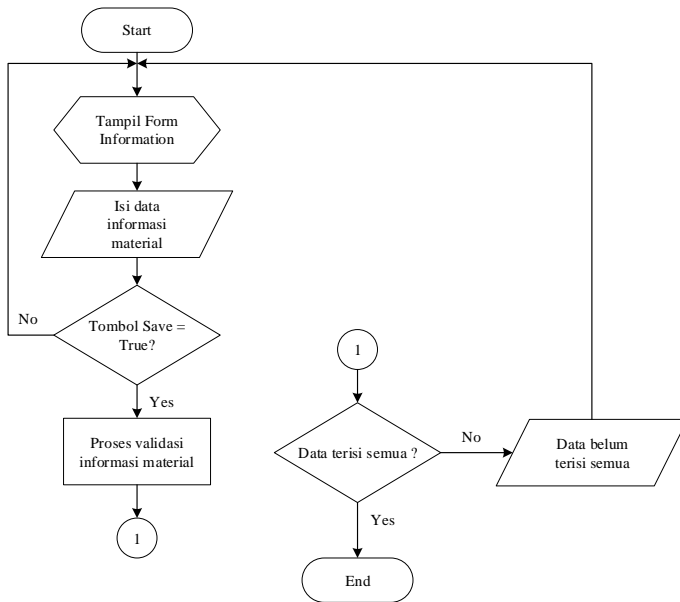
pengguna dapat menekan tombol *result* dan semua informasi hasil monitoring akan ditampilkan pada form ini.



Gambar 3.5 Flowchart Form Main Menu

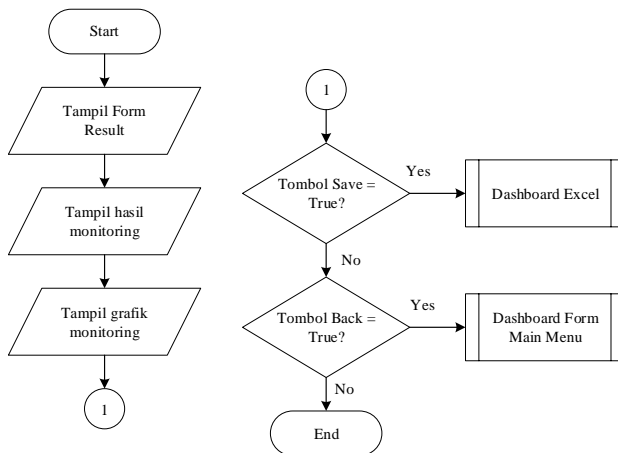
3.2.1.4 Flowchart Form Information

Gambar 3.6 menjelaskan *flowchart* form informasi. Pada form ini berisi tentang data informasi material awal yang harus diisi sebelum melakukan proses pengukuran atau monitoring. Data ini nantinya berfungsi untuk laporan hasil akhir saat proses pengukuran selesai. Jika data yang diisi belum lengkap, maka akan muncul pemberitahuan untuk mengisi data yang belum terisi, jika sudah terisi semua data dapat disimpan dan dilanjutkan ke proses berikutnya.



Gambar 3. 6 *Flowchart Form Information*

3.2.1.5 Flowchart Form Result

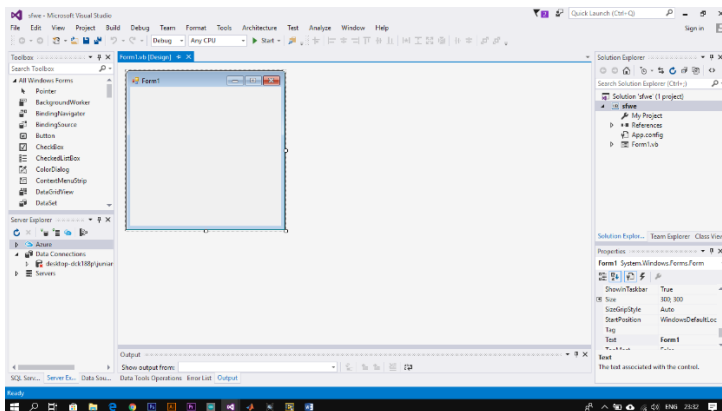


Gambar 3. 7 *Flowchart Form Result*

Gambar 3.7 menjelaskan *flowchart form result*. Pada form ini semua data material yang sudah dimasukkan, grafik hasil pengukuran, dan hasil pengukuran akan ditampilkan. Form ini berfungsi untuk pengguna agar lebih mudah dalam menganalisa material yang telah diuji. Lalu pengguna dapat menyimpan data monitoring tersebut dalam bentuk excel.

3.2.1.6 Tampilan HMI Pada Aplikasi UTM

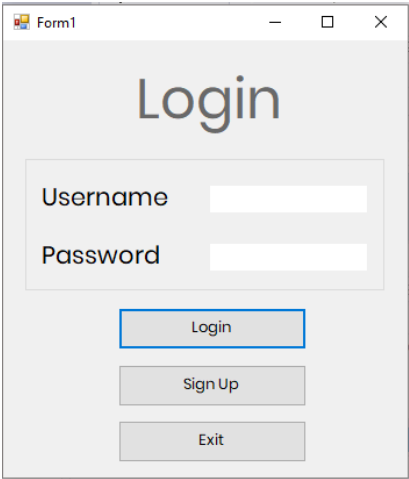
Pembuatan aplikasi ini bertujuan untuk memudahkan pengguna dalam memonitoring hasil pengukuran mesin. *Software* yang digunakan untuk perancangan aplikasi HMI menggunakan *Visual Studio 2015*. Dimana dalam pembuatan program *project* yang digunakan adalah *Visual Basic*. Selanjutnya akan tampil *Design Form* kosong seperti pada Gambar 3.8 dan nantinya digunakan untuk menyusun komponen dan mendesain HMI yang akan dibuat. Letak dari komponen-komponen tersebut berada pada sisi kiri dan untuk menambahkannya pada form kosong, pengguna cukup melakukan *drag and drop* komponen yang sudah dipilih ke dalam form kosong. Untuk penggunaan fungsi dari komponen tersebut tergantung dengan kebutuhan. Pada Gambar 3.8 merupakan tampilan awal saat membuat *project* baru di *Visual Basic*.



Gambar 3. 8 Tampilan Project Baru

Pada perancangan *software* ini terdapat 5 bagian tampilan yang memiliki peran penting saat aplikasi digunakan. Bagian



tampilan itu diantaranya yaitu form *login*, form sign up, form main menu, form information, dan form result. Untuk form *login* ini pengguna diharuskan mengisi Username dan Password yang sudah dimiliki agar dapat mengakses aplikasi lebih lanjut, maka yang komponen dibutuhkan untuk mengisikan username dan password adalah *textbox*. Berikut pada Gambar 3.9 merupakan tampilan dari form *login* yang sudah dibuat.

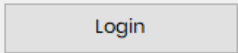
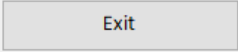
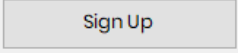


Gambar 3. 9 Tampilan *Form Login*

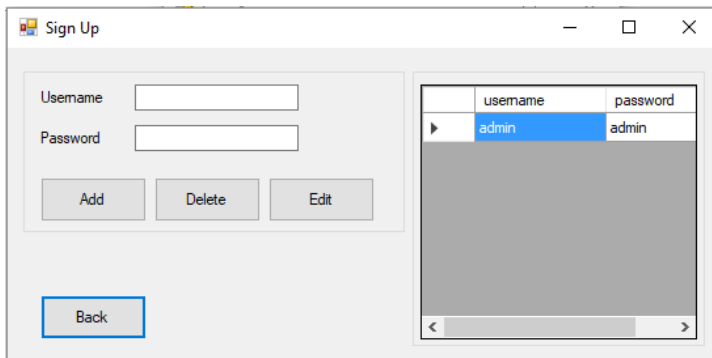
Dapat dilihat pada Gambar 3.9 merupakan bentuk tampilan dari HMI form *Login* yang digunakan untuk pengaman, agar tidak semua orang dapat mengakses aplikasi ini dengan bebas. Jika pengguna belum memiliki akun untuk mengakses software ini, pengguna juga dapat membuat akun baru dengan menekan tombol *Sign Up*. Berikut penjelasan tampilan HMI pada form *login* seperti pada Tabel 3.1.

Tabel 3. 1 Komponen Tampilan HMI Pada *Form Login*

Ikon		Keterangan
		Masukan <i>Username</i> pada <i>Textbox</i>
		Masukan <i>Password</i> pada <i>Textbox</i>

Ikon	Keterangan
	Tombol <i>Login</i>
	Tombol keluar aplikasi
	Masuk <i>form Sign Up</i>

Dapat dilihat pada Tabel 3.1 untuk kolom pertama merupakan komponen yang terdapat pada form *login*, dan untuk kolom kedua yaitu kegunaan komponen dalam form tersebut. Kemudian untuk membuat akun baru, pengguna dapat menekan tombol *Sign Up*. Pada form *Sign Up* ini pengguna dapat membuat akun baru, menghapus akun, dan memperbarui password akun. Setelah form *Sign Up* muncul pengguna diharuskan mengisi username dan password pada textbox kemudian tekan tombol *Add* untuk menambahkan akun baru. Jika ingin menghapus akun yang sudah ada, pengguna harus memilih akun yang akan dihapus pada tabel kemudian tekan tombol *Delete*. Dan untuk memperbarui password akun pilih akun yang akan diperbarui pada table kemudian ganti password pada textbox setelah itu tekan edit. Berikut merupakan gambar tampilan form *Sign Up* pada Gambar 3.10.



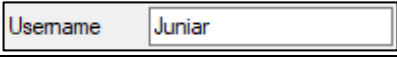
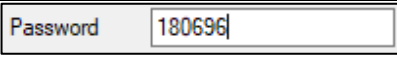
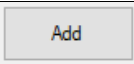
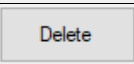
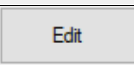
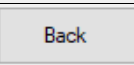
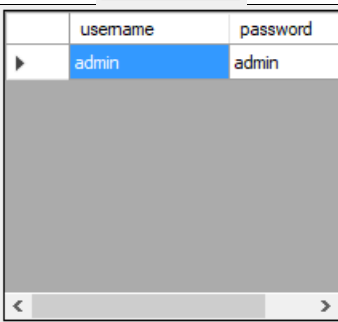
username	password
admin	admin

Gambar 3. 10 Tampilan *Form Sign Up*

Pada Gambar 3.10 merupakan bentuk tampilan HMI untuk form *Sign Up* yang digunakan untuk membuat akun baru. Form ini

memiliki tiga pilihan tombol dan dua *textbox* untuk mengisi username dan password. Berikut adalah penjelasan tiap komponen pada form *Sign Up* seperti pada Tabel 3.2. Pada tabel tersebut terdapat dua kolom, untuk kolom pertama merupakan komponen-komponen yang terdapat pada form *sign up*. Untuk kolom kedua adalah penjelasan dari kegunaan tiap komponen yang ada pada form *sign up*.

Tabel 3. 2 Komponen Tampilan HMI Pada *Form Sign Up*

Ikon	Keterangan
	Masukkan <i>Username</i>
	Masukkan <i>Password</i>
	Tombol untuk menambah akun
	Tombol untuk menghapus akun
	Tombol untuk memperbarui akun
	Tombol untuk kembali ke <i>Login</i>
	Tabel daftar akun yang terdaftar

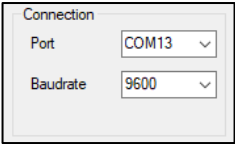

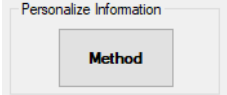
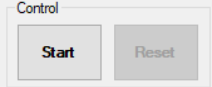

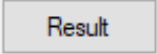
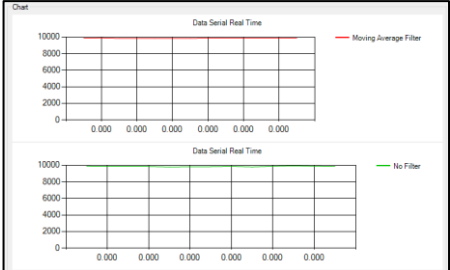
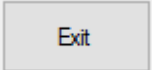
Setelah proses *login* selesai maka pengguna akan masuk ke tampilan *form Main menu*. Pada form ini terdapat bagian utama yang harus di setting yaitu berisi konektivitas antara komputer dengan mikrokontroler. Pengguna diharuskan memilih port dan baudrate sesuai dengan mikrokontroler yang sedang digunakan. Selain itu, juga terdapat tampilan monitoring pengukuran yaitu berupa grafik

hasil pengukuran, data sumbu x, data sumbu y, titik luluh, dan titik maksimal. Sebelum pengguna dapat melakukan monitoring pengukuran, pengguna harus mengisi data informasi objek yang akan diukur dengan menekan tombol *Method*, dan setelah data terisi semua tekan tombol *Start* untuk mulai melakukan monitoring pengukuran dan setelah mesin selesai melakukan pengukuran tekan tombol *Stop*. Jika monitoring pengukuran selesai, pengguna dapat melihat hasil pengukuran beserta grafik dengan menekan tombol *Result*. Tampilan form *Main menu* dari aplikasi HMI ini ditunjukkan pada Gambar 3.11

Gambar 3. 11 Tampilan *Form Main Menu*

Dapat dilihat pada Gambar 3.11 merupakan tampilan HMI form *Main menu*, pada form ini terdapat beberapa komponen-komponen penyusun form yang memiliki peran penting dalam melakukan proses monitoring. Berikut penjelasan komponen-komponen untuk tampilan HMI form *Main Menu* dapat dilihat pada Tabel 3.3. Pada tabel tersebut terdapat dua kolom, untuk kolom pertama berisi komponen-komponen apa saja yang terdapat pada form *Main Menu*. Kemudian pada kolom kedua terdapat penjelasan dari kegunaan tiap komponen dalam form ini.

Tabel 3. 3 Komponen Tampilan HMI Pada *Form Main Menu*

Ikon	Keterangan
	<p><i>GroupBox</i> untuk setting koneksi</p>
	<p>Tombol untuk koneksi</p>
	<p>Tombol masuk form Informasi</p>
	<p>Tombol untuk control monitoring</p>
	<p><i>GroupBox</i> untuk menampilkan data hasil monitoring</p>
	<p>Tombol masuk form Hasil</p>
	<p>Letak monitoring grafik pengukuran</p>
	<p>Tombol untuk keluar dari aplikasi</p>

Sebelum melakukan monitoring pengukuran, pengguna harus mengisi form Informasi terlebih dahulu dengan menekan tombol *Method*. Setelah form muncul, pengguna mengisi data awal dari objek yang akan diukur pada *textbox* yang telah disediakan. Jika semua informasi yang diperlukan sudah terisi tekan tombol *Save* dan

kembali ke form *Main menu* untuk mulai melakukan monitoring. Berikut adalah tampilan dari form Informasi dapat dilihat pada Gambar 3.12.

Gambar 3. 12 Tampilan *Form Information*

Dan berikut adalah penjelasan komponen-komponen yang ada di form Informasi dapat dilihat pada Tabel 3.4. Pada tabel tersebut terdapat dua kolom, untuk kolom pertama merupakan komponen-komponen yang terdapat pada form *Information*. Dan kolom kedua adalah penjelasan atau fungsi dari komponen yang ada pada form tersebut.

Tabel 3. 4 Komponen Tampilan HMI Pada *Form Information*

Ikon	Keterangan
<div> <div>Fill Data</div> <div> <div>Order*</div> <div></div> <div>Lo (mm)*</div> <div></div> </div> <div> <div>Project*</div> <div></div> <div>d (mm)*</div> <div></div> </div> <div> <div>Date</div> <div>Minggu, 15 Juli ▾</div> <div>Diameter*</div> <div></div> </div> <div> <div>Tester*</div> <div></div> <div>Long*</div> <div></div> </div> <div> <div>Sample Code*</div> <div></div> <div>Wide*</div> <div></div> </div> </div>	<div> <div>GroupBox</div> <div>untuk</div> <div>mengisi</div> <div>data</div> <div>informasi</div> </div>
<div> <div>Back</div> </div>	<div> <div>Tombol</div> <div>untuk</div> <div>kembali</div> <div>ke</div> <div>Main</div> <div>menu</div> </div>
<div> <div>Save</div> </div>	<div> <div>Tombol</div> <div>untuk</div> <div>menyimpan</div> <div>data</div> <div>informasi</div> </div>


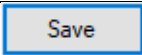
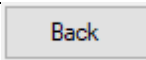
Kemudian proses yang terakhir yaitu melihat hasil akhir pengukuran dengan menekan tombol *Result* pada form *Main menu*. Setelah form muncul data hasil monitoring akan ditampilkan secara otomatis. Data yang ditampilkan yaitu berupa grafik dan data Informasi yang sudah disimpan. Berikut ini yaitu tampilan dari form *Result* dapat dilihat pada Gambar 3.13.

Gambar 3. 13 Tampilan *Form Result*

Berikut ini merupakan penjelasan komponen pada tampilan form *Result* dapat dilihat pada Tabel 3.5. Pada tabel tersebut terdapat dua kolom, untuk kolom pertama merupakan komponen-komponen yang terdapat pada form *result*. Untuk kolom kedua adalah penjelasan atau kegunaan tiap komponen yang ada pada form *result*.

Tabel 3. 5 Komponen Tampilan HMI Pada *Form Result*

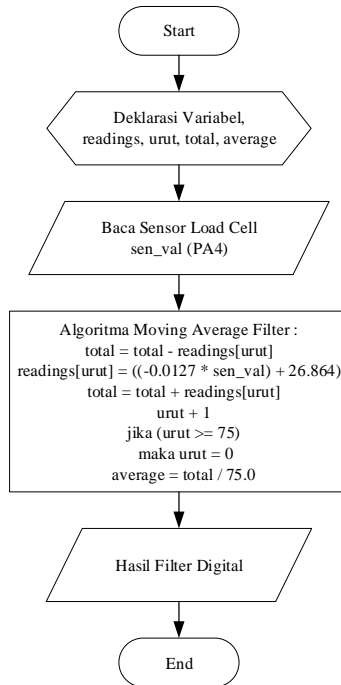
Ikon	Keterangan
<div> <div>Data</div> <div>Order :</div> <div>Project :</div> <div> <div>Date</div> <div>00.00.00</div> <div>Pu (kgf)</div> <div></div> </div> <div> <div>Test Machine</div> <div></div> <div>Delta (mm)</div> <div></div> </div> <div> <div>Tester</div> <div></div> </div> <div> <div>Sample Code</div> <div></div> </div> <div> <div>Sample Form</div> <div></div> </div> <div> <div>Lo (mm)</div> <div></div> </div> <div> <div>d (mm)</div> <div></div> </div> </div>	<div> <div>GroupBox</div> <div>tampilan data hasil monitoring</div> </div>

Ikon	Keterangan
	Grafik hasil monitoring.
	Tombol untuk menyimpan file baru
	Tombol untuk kembali ke <i>Main menu</i>

3.2.2 Perancangan Program Mikrokontroler

Perancangan program pada mikrokontroler ini berkaitan dengan pembuatan filter digital untuk data dari sensor yang masuk ke mikrokontroler dan untuk membaca sinyal analog yang kemudian diubah menjadi sinyal digital dan dikirimkan ke komputer melalui komunikasi serial. Pembuatan filter digital pada mikrokontroler ini berfungsi untuk mengurangi *random noise* pada data sensor yang diterima mikrokontroler. Dalam pemrograman mikrokontroler ini, rumus *Moving Average Filter* yang sudah ada dimasukkan ke dalam program berupa kode program.

Sebelum membuat program pada mikrokontroler dilakukan pembuatan *flowchart* untuk proses *Moving Average Filter*. Pada Gambar 3.14 menjelaskan *flowchart Moving Average Filter*. Saat proses ini berlangsung data yang masuk ke mikrokontroler akan di inisialisasi terlebih dahulu ke dalam sebuah variabel. Kemudian variabel tersebut yang nantinya akan diolah dalam fungsi *Moving Average Filter* pada mikrokontroler. Setelah itu data yang sudah di filter langsung dikirim ke komputer untuk dilakukan proses monitoring.



Gambar 3. 14 Flowchart Perancangan *Moving Average Filter*

Untuk langkah awal dalam pembuatan suatu program mikrokontroler yaitu mendeklarasikan variabel dan tipe data yang akan digunakan. Variabel dan tipe data ini berfungsi untuk memudahkan dalam menempatkan suatu data, agar data tersebut dapat di program dan diolah kembali menjadi data yang lain. Adapun deklarasi variable yang sudah dibuat dapat dilihat pada Gambar 3.15.

```

double readings[75];
int urut = 0;
double total = 0;
double average = 0;
  
```

Gambar 3. 15 Listing Program Deklarasi Variabel dan Tipe Data

Proses selanjutnya yaitu pengaturan pin mikrokontroler pada *void setup*, pada bagian ini *void setup* digunakan untuk menginisialisasi pin digital yang akan kita gunakan sebagai *input* atau *output*, serta setting baudrate untuk mikrokontroler. Pada perancangan program yang sudah dibuat, pengenalan pengiriman komunikasi serial menggunakan baudrate 9600. Kemudian untuk pengenalan pin yang mendapatkan sinyal analog yaitu pin A4 untuk sensor load cell dan pin A5 untuk sensor presisi. Dan berikut merupakan inisialisasi *void setup* yang sudah dibuat dapat dilihat pada Gambar 3.16.

```
void setup() {  
  Serial.begin(9600);  
  pinMode(PC13, OUTPUT);  
  pinMode(PA4, INPUT);  
  pinMode(PA5, INPUT);  
  pinMode(pinA, INPUT_PULLUP);  
  pinMode(pinB, INPUT_PULLUP);  
  digitalWrite(PC13, LOW);  
  
  for (int i = 0; i < 75; i++) {  
    readings[i] = 0;  
  }  
}
```

Gambar 3. 16 Listing Program Inisialisasi pada *Void Setup*

Pembuatan fungsi Moving Average ini digunakan untuk filter data dari sensor load cell. Semua data dari sensor load cell yang masuk ke mikrokontroler akan langsung diproses pada fungsi ini sebelum nantinya dikirimkan ke komputer. Fungsi ini diletakkan pada Void Loop dalam struktur program, agar fungsi ini dapat diproses saat program dijalankan. Pembuatan program ini didapat dari rumus Moving Average Filter yang sudah ada. Cara kerja dari filter ini yaitu seluruh data yang sudah dibaca oleh mikrokontroler akan ditempatkan pada sebuah variabel bernama “*sen_val*”. Lalu dibuat variabel baru untuk menempatkan data lama yang sudah terbaca oleh variabel sebelumnya. Kemudian dibuat perulangan untuk mengambil tiap 75 data yang terbaca untuk dijumlahkan dan hasilnya dibagi sebanyak data yang sudah diambil. Proses tersebut akan berulang terus menerus dengan mengambil 75 data selanjutnya selama mikrokontroler menerima data dari sensor. Pada gambar 3.17

merupakan program yang sudah dibuat untuk *Moving Average Filter*.

```
total = total - readings[urut];  
int sen_val = analogRead(PA4);  
readings[urut] = ((-8.3065 * sen_val) + 16035);  
total = total + readings[urut];  
urut++;  
if (urut >= 75) {  
    urut = 0;  
}  
average = total / 75.0;
```

Gambar 3. 17 Listing Program Fungsi *Moving Average Filter*

Proses akhir dalam pembuatan *Moving Average Filter* yaitu pengiriman data yang sudah di filter ke komputer untuk dilakukan monitoring. Program tersebut diletakkan pada *Void Loop* yang berfungsi menampilkan parameter hasil pembacaan sensor. Dengan menggunakan fungsi *Serial.print* yang artinya menampilkan parameter yang terbaca oleh mikrokontroler. Pada fungsi ini nantinya akan menampilkan 2 parameter dari sensor yaitu *Serial.print(hasil_mv)*; menampilkan hasil dari filter digital dan *Serial.print(sensor2)*; menampilkan data dari sensor. Berikut merupakan program untuk mengirimkan parameter dapat dilihat pada Gambar 3.18.

```
Serial.print("|");  
Serial.print(jarak, 3);  
Serial.print("|");  
Serial.print(readings[urut]);  
Serial.print("|");  
Serial.print(average);  
Serial.println("|");
```

Gambar 3. 18 Listing Program Pengiriman Data Mikrokontroler dengan Komputer

-----Halaman ini sengaja dikosongkan-----

BAB IV

PENGUJIAN DAN ANALISA DATA

Untuk mengetahui apakah tujuan-tujuan dari pengerjaan Tugas Akhir ini telah tercapai atau belum, maka perlu dilakukan pengujian, pengukuran dan analisa terhadap alat yang sudah ada, kemudian diintegrasikan untuk melihat kondisi yang sama antara *hardware* dan *software*. Hal ini dilakukan dengan tujuan untuk mengetahui performansi sistem secara keseluruhan. Mekanisme kerja sistem HMI pada UTM berdasarkan tampilan data pengukuran beban dan perubahan posisi saat proses pengukuran berlangsung. Data yang terbaca oleh mikrokontroler akan diolah dan hasilnya akan dikirimkan ke komputer untuk ditampilkan pada HMI yang sudah dibuat. Setelah itu dilakukan proses evaluasi agar dapat membawa alat ini ke arah yang lebih baik. Pengujian yang akan dilakukan untuk sistem HMI pada UTM yaitu meliputi:

1. Pengujian penerimaan data tekanan dari *Load Cell*.
2. Pengujian penerimaan data perubahan posisi.
3. Pengujian filter digital *Moving Average* pada data tekanan *Load Cell*.

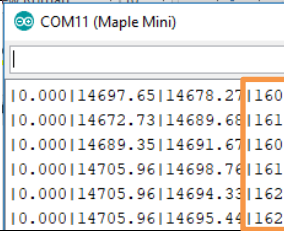

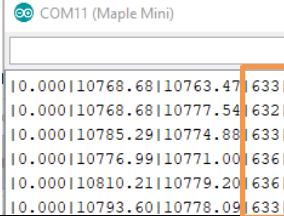
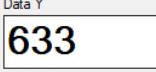
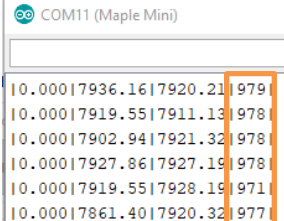
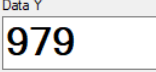
4.1 Pengujian Penerimaan Data Tekanan Dari Sensor *Load Cell*

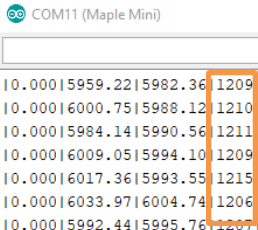
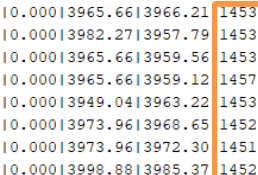
Tujuan dari pengujian penerimaan data tekanan dari sensor load cell ini digunakan untuk mengetahui data yang diterima oleh HMI sama dengan data yang dikirimkan oleh mikrokontroler. Pengujian ini dilakukan dengan mengirimkan data dari mikrokontroler (master) ke laptop/komputer (*slave*) melalui komunikasi serial. Pengujian dilakukan dengan membandingkan keluaran dari mikrokontroler menggunakan Serial Monitor Arduino dengan keluaran data serial menggunakan tampilan HMI.

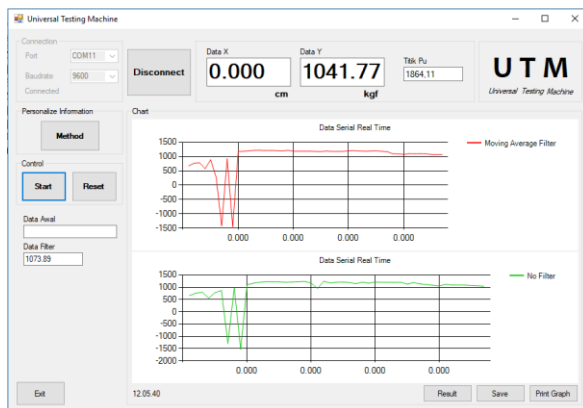
Pertama yang harus dilakukan yaitu catat beberapa data tetap yang terbaca pada mikrokontroler pada serial monitor. Setelah itu jalankan program HMI dan koneksikan dengan mikrokontroler, data tetap yang sudah dikirimkan mikrokontroler ke HMI akan tampil, lalu bandingkan data tetap yang sudah dicatat dengan tampilan pada HMI. Pada pengujian ini mengambil 5 contoh data pada Tabel 4.1 berikut ini. Pada tabel tersebut terdapat tiga kolom yang menjelaskan hasil pengujian yang sudah dilakukan. Kolom pertama adalah data

yang akan dikirimkan mikrokontroler. Data ini didapatkan dari hasil pembacaan sensor *load cell*. Kemudian untuk melihat data yang dikirimkan benar yaitu dengan membuka serial monitor yang dapat dilihat pada kolom kedua. Setelah itu data pada serial monitor dibandingkan dengan data yang terbaca pada HMI yang terdapat pada kolom ketiga.

Tabel 4. 1 Hasil Pengujian Penerimaan Data *Load Cell*

No	Data dikirim dari mikrokontroler (ADC)	Data yang diterima oleh PC	
		Serial Monitor	HMI (Human Machine Interface)
1.	160		
2.	633		
3.	979		

No .	Data dikirim dari mikrokontroler (ADC)	Data yang diterima oleh PC	
		Serial Monitor	HMI (Human Machine Interface)
4.	1209		Data Y 1209
5.	1453		Data Y 1453



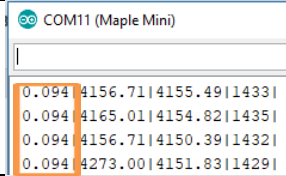
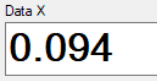
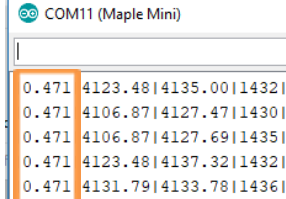

Gambar 4. 1 Tampilan pada HMI Saat Proses Pengujian Load Cell Berlangsung

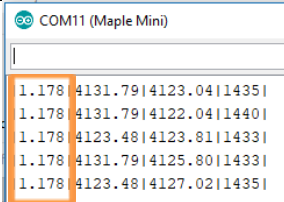
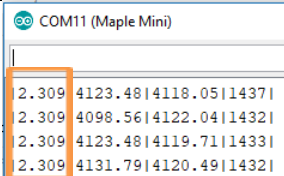
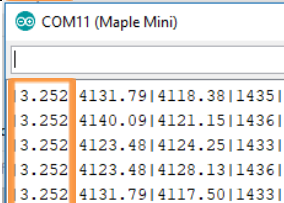
Pada Gambar 4.1 adalah tampilan pada HMI saat proses pengujian atau monitoring penerimaan data *load cell* berlangsung. Dapat dilihat pada kolom Data Y menampilkan angka yang terbaca dari mikrokontroler. Kemudian dibawah kolom terdapat gambar grafik yang menampilkan data perubahan dari sensor *load cell* yang terbaca pada sumbu y.

4.2 Pengujian Penerimaan Data Perubahan Posisi

Untuk pengujian kedua ini bertujuan untuk mengetahui perubahan posisi yang terbaca pada mikrokontroler sama dengan data yang tampil pada HMI. Langkah-langkah yang dilakukan dalam pengujian ini sama seperti pengujian sebelumnya. Pertama catat data tetap perubahan posisi yang terbaca pada serial monitor arduino, kemudian jalankan program HMI dan koneksikan dengan mikrokontroler, maka data tetap yang sudah dikirimkan pada mikrokontroler akan tampil pada HMI, kemudian bandingkan data serial monitor dengan yang tampil di HMI. Pada pengujian ini mengambil 5 contoh data pada Tabel 4.2 berikut ini.

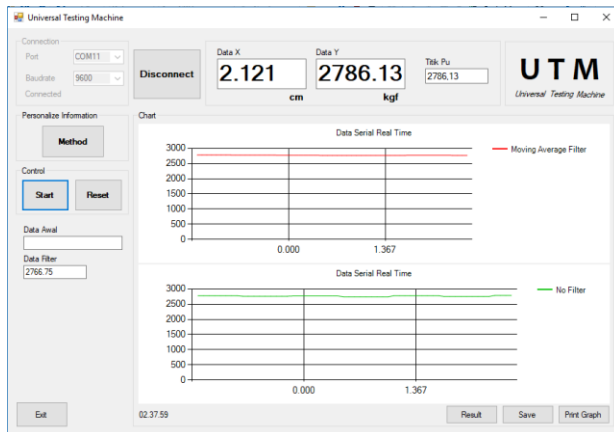
Tabel 4. 2 Hasil Pengujian Penerimaan Data Perubahan Posisi

No .	Data dikirim dari mikrokontrol er (mm)	Data yang diterima oleh PC	
		Serial Monitor	HMI (Human Machine Interface)
1.	0,094	 <pre> COM11 (Maple Mini) 0.094 4156.71 4155.49 1433 0.094 4165.01 4154.82 1435 0.094 4156.71 4150.39 1432 0.094 4273.00 4151.83 1429 </pre>	
2.	0,471	 <pre> COM11 (Maple Mini) 0.471 4123.48 4135.00 1432 0.471 4106.87 4127.47 1430 0.471 4106.87 4127.69 1435 0.471 4123.48 4137.32 1432 0.471 4131.79 4133.78 1436 </pre>	

No .	Data dikirim dari mikrokontroler (mm)	Data yang diterima oleh PC	
		Serial Monitor	HMI (Human Machine Interface)
3.	1,178	 <pre> COM11 (Maple Mini) 1.178 4131.79 4123.04 1435 1.178 4131.79 4122.04 1440 1.178 4123.48 4123.81 1433 1.178 4131.79 4125.80 1433 1.178 4123.48 4127.02 1435 </pre>	Data X 1.178
4.	2,309	 <pre> COM11 (Maple Mini) 2.309 4123.48 4118.05 1437 2.309 4098.56 4122.04 1432 2.309 4123.48 4119.71 1433 2.309 4131.79 4120.49 1432 </pre>	Data X 2.309
5.	3,252	 <pre> COM11 (Maple Mini) 3.252 4131.79 4118.38 1435 3.252 4140.09 4121.15 1436 3.252 4123.48 4124.25 1433 3.252 4123.48 4128.13 1436 3.252 4131.79 4117.50 1433 </pre>	Data X 3.252

Pada tabel tersebut terdapat tiga kolom yang menjelaskan hasil pengujian yang telah dilakukan. Kolom pertama adalah data yang akan dikirimkan mikrokontroler. Data ini didapatkan dari hasil pembacaan perubahan posisi. Kemudian untuk melihat data yang dikirimkan benar yaitu dengan membuka serial monitor pada arduino IDE yang dapat dilihat pada kolom kedua. Setelah itu data pada serial monitor dibandingkan dengan data yang terbaca pada tampilan HMI saat dijalankan yang terdapat pada kolom ketiga.

Berikut ini adalah tampilan pada tampilan HMI saat proses pengujian penerimaan data perubahan posisi berlangsung dapat dilihat pada Gambar 4.2.

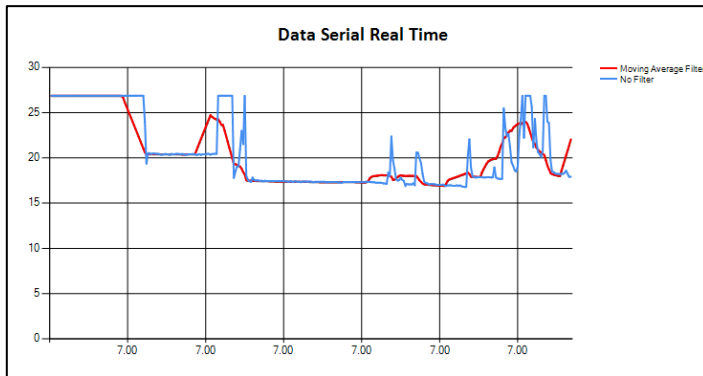


Gambar 4. 2 Tampilan pada HMI Saat Proses Pengujian Perubahan Posisi Berlangsung

Gambar 4.2 adalah tampilan pada HMI saat proses pengujian atau monitoring penerimaan data perubahan posisi berlangsung. Dapat dilihat pada kolom Data X menampilkan angka yang terbaca dari mikrokontroler. Kemudian dibawah kolom terdapat gambar grafik yang menampilkan data perubahan posisi yang terbaca pada sumbu x.

4.3 Pengujian Filter Digital *Moving Average*

Tujuan dari pengujian filter digital moving average pada program mikrokontroler adalah untuk mengetahui apakah sinyal yang dihasilkan dapat mereduksi noise. Saat proses pengukuran berlangsung, data dari sensor yang sudah terbaca pada mikrokontroler akan diolah dengan persamaan *moving average*, yang kemudian persamaan tersebut diimplementasikan menjadi sebuah kode program yang dapat mereduksi noise saat pengiriman data berlangsung. Setelah itu mikrokontroler akan mengirimkan data hasil filter dan data mentah sebelum di filter pada HMI, yang nantinya data tersebut akan diubah menjadi grafik agar dapat melihat perbandingan data yang menggunakan filter digital dan tanpa filter digital. Hasil sinyal yang dihasilkan oleh HMI ditunjukkan oleh Gambar 4.3.



Gambar 4. 3 Tampilan Grafik Perbandingan Data dengan Filter Moving Average dan Data Tanpa Filter.

Pada Gambar 4.3 dapat dilihat bahwa terdapat 2 grafik yang terbaca oleh HMI. Untuk grafik dengan warna garis biru adalah hasil grafik dari data mentah yang terbaca oleh mikrokontroler. Kemudian dari data mentah yang didapat akan dibandingkan dengan data yang sudah melalui proses filter digital. Untuk data grafik dengan warna garis merah merupakan hasil grafik dari data yang menggunakan filter digital *Moving Average*.

-----Halaman ini sengaja dikosongkan-----

BAB V

PENUTUP

Bab penutup ini berisi tentang kesimpulan-kesimpulan yang didapatkan selama proses pembuatan Tugas Akhir Sistem *Human Machine Interface* Pada *Universal Testing Machine*, beserta saran-saran perbaikan dan pengembangan untuk kedepannya.

5.1 Kesimpulan

Setelah melakukan pengujian serta analisa data dari Tugas Akhir yang berjudul Sistem *Human Machine Interface* Pada *Universal Testing Machine*. Maka dapat diambil kesimpulan sebagai berikut:

1. HMI yang dibuat dengan menggunakan *software Visual Basic* sudah menunjukkan hasil yang real time dan mudah dalam pengaplikasiannya. Sehingga memudahkan pengguna dalam memonitoring mesin UTM.
2. Filter digital dengan metode *Moving Average* dapat meredam noise dan memiliki hasil keluaran yang bagus. Hasil pada grafik perbandingan juga menunjukkan mendekati hasil beban load cell dan menghasilkan grafik yang halus.

5.2 Saran

Untuk pengembangan dan penyempurnaan dari Pembuatan Perangkat Lunak Sistem *Human Machine Interface* Pada *Universal Testing Machine* ini maka diberikan beberapa saran sebagai berikut:

1. Sebaiknya HMI dengan *software Visual Basic* ini ditambahkan program untuk menghapus data serial buffer saat proses penerimaan data berlangsung, agar banyaknya data yang terbaca oleh HMI tidak dapat menghambat proses penerimaan data dan membuat HMI tidak dapat membaca data yang diterima.
2. Pembuatan HMI untuk alat pengukuran sebaiknya tidak menggunakan delay atau nilai delay yang dipakai seminimal mungkin, agar tidak ada data yang hilang saat proses pengukuran berlangsung.

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

- [1] Rakhmawati Amalia. 2011. <http://www.Sites.google.com/site/calibrationconsultancy> (Diakses pada 20 April, 2018)
- [2] S.H. Iimin. 2012. Rancang Bangun Sistem Human Machine Interface (HMI) Pada Miniplant Pembangkit Listrik Tenaga Mikrohidro Skala Laboratorium. Tugas Akhir. Institut Teknologi Sepuluh Nopember. Surabaya.
- [3] Haryanto Heri, Hidayat Sarif. 2012. Perancangan HMI (*Human Machine Interface*) Untuk Pengendalian Kecepatan Motor DC. Universitas Sultan Ageng Tirtayasa. Banten.
- [4] Suyadi. 2012. Komunikasi Serial dan Port Serial (COM). Universitas Muhammadiyah Surakarta. Surakarta.
- [5] Sianipar R.H. 2014. Pemrograman Visual Basic .NET. Bandung : Informatika Bandung.
- [6] Hermawi Adrianto. 2007. Aplikasi Moving Average Filter Pada Teknologi Enkripsi. Volume 9. Nomor 1.
- [7] Hariyanto, Didik. 2009. Teknik Antarmuka ADC. Staff Site Universitas Negeri Yogyakarta. <http://directlinksearch.com/dir/teknikantarmuka> (Diakses pada 11 April, 2018)
- [8] Ddrown. 2009. <http://www.stm32duino.com/viewtopic.php?t=844> (Diakses pada 16 April, 2018)
- [9] Burch, J.G., 1992, *System, Analysis, Design, and Implementation*, Boyd & Fraser Publishing Company.

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN A

LISTING PROGRAM

A. *Listing Program Mikrokontroler*

```
double readings[75];
int urut = 0;
double total = 0;
double average = 0;
unsigned long timeold1;
byte data;
float sensor1, sensor2;
float data1, data2;
int pinA = PB8;
int pinB = PB9;
int counter = 0;
int counterPinlast = LOW;
int counterAlast = LOW;
int counterBlast = LOW;
int n = LOW;
int A = LOW;
int B = LOW;
const float pi = 3.1415926536;
const float R = 6;
const int N = 400;
float jarak;

void setup() {
  Serial.begin(9600);
  pinMode(PA4, INPUT);
  pinMode(pinA, INPUT_PULLUP);
  pinMode(pinB, INPUT_PULLUP);

  for (int i = 0; i < 75; i++) {
    readings[i] = 0;
  }
}

void loop() {
```

```

n = digitalRead(pinA);
if ((counterPinlast != LOW) && (n != HIGH)) {
    if (digitalRead(pinB) != LOW) {
        counter--;
    } else {
        counter++;
    }
    jarak = (pi * R) * counter / N;
}

total = total - readings[urut];
int sen_val = analogRead(PA4);
readings[urut] = ((-0.0127 * sen_val) + 26.864);
total = total + readings[urut];
urut++;
if (urut >= 75) {
    urut = 0;
}
average = total / 75.0;

if (Serial.available() > 0)
{
    data = Serial.read();
}
switch (data) {
case 97:
    if (millis() - timeold1 >= 10) {
        Serial.print("|");
        Serial.print(jarak, 3);
        Serial.print("|");
        Serial.print(readings[urut]);
        Serial.print("|");
        Serial.print(average);
        Serial.println("|");
        timeold1 = millis();
    }
    break;
case 98:
    nvic_sys_reset();

```

```

        break;
    }
    counterPinlast = n;
}

```

B. *Listing Program Visual Basic Form Login*

```

Imports System.Data.SqlClient
Public Class Form1
    Private Sub btlogin_Click(sender As Object, e
As EventArgs) Handles btlogin.Click
        Dim conString As String = "Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|Dat
aDirectory|\newlogin.mdf;Integrated
Security=True;Connect Timeout=30"
        Dim con As New SqlConnection(conString)
        Dim cmdText As String = "SELECT * FROM
LOGIN"
        Dim cmd As New SqlCommand(cmdText, con)
        Dim da As New SqlDataAdapter(cmd)
        Dim dt As New DataTable

        con.Open()
        da.Fill(dt)
        con.Close()

        Dim dr As DataRow
        Dim dc As DataColumn
        Dim arr(1, 3) As Object

        For Each dr In dt.Rows
            Dim ctr As Integer = 0
            For Each dc In dt.Columns
                arr(0, ctr) = dr(dc.ColumnName)
                ctr = ctr + 1
            Next

            If tusername.Text = arr(0, 0) And
tpassword.Text = arr(0, 1) Then

```

```

        MessageBox.Show("Berhasil", "LOG
IN", MessageBoxButtons.OK)
        Me.Hide()
        fMenu.Show()
        Exit Sub
    End If
Next
MsgBox("Sorry, username or password not
found", MsgBoxStyle.OkOnly, "Invalid")
End Sub
Private Sub btdaftar_Click(sender As Object, e
As EventArgs) Handles btdaftar.Click
    fsignup.Show()
End Sub
Private Sub btkeluar_Click(sender As Object, e
As EventArgs) Handles btkeluar.Click
    Close()
End Sub
End Class

```

C. *Listing Program Visual Basic Form Sign Up*

```

Imports System.Data.SqlClient

Public Class fsignup
    Private Sub
LoginBindingNavigatorSaveItem_Click(sender As
Object, e As EventArgs)
        Me.Validate()
        Me.LoginBindingSource.EndEdit()

Me.TableAdapterManager.UpdateAll(Me.NewloginDataSet
)
    End Sub

    Private Function TableAdapterManager() As
Object
        Throw New NotImplementedException()
    End Function

```



```

        Private Sub login_Load(sender As Object, e As
EventArgs) Handles MyBase.Load

Me.LoginTableAdapter.Fill(Me.NewloginDataSet.login)
End Sub

        Private Sub bkembali_Click(sender As Object, e
As EventArgs) Handles bkembali.Click
            Me.Close()
        End Sub

        Private Sub btambah_Click(sender As Object, e
As EventArgs) Handles btambah.Click
            Dim username As String = usrname.Text
            Dim password As String = psswd.Text

            Dim conString As String = "Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|Dat
aDirectory|\newlogin.mdf;Integrated
Security=True;Connect Timeout=30"
            Dim con As New SqlConnection(conString)
            Dim cmdText As String = "INSERT INTO login
VALUES (@username, @password)"
            Dim cmd As New SqlCommand(cmdText, con)

            cmd.Parameters.AddWithValue("@username",
username)
            cmd.Parameters.AddWithValue("@password",
password)

            Try
                con.Open()
                cmd.ExecuteNonQuery()
                MessageBox.Show("Berhasil", "Daftar",
MessageBoxButtons.OK)
                LoginDataGridView.DataSource = Nothing
                LoginDataGridView.Refresh()
                Dim str As String = "SELECT * FROM
login"
                cmd = New SqlCommand(str, con)

```

```

        Dim da As New SqlDataAdapter(cmd)
        Dim dt As New DataTable
        da.Fill(dt)
        LoginDataGridView.DataSource = dt
    Catch ex As Exception
        MsgBox("Sorry, username sama",
        MsgBoxStyle.OkOnly, "Invalid")
    Finally
        con.Close()
    End Try
End Sub

Private Sub bhapus_Click(sender As Object, e As
EventArgs) Handles bhapus.Click
    If username.Text = "" Or psswd.Text = ""
Then
        MsgBox("Pilih Username yang akan
dihapus!")
    ElseIf MsgBox("Apakah anda yakin akan
menghapus username ini?") = MsgBoxResult.Ok Then
        Dim conString As String = "Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|Dat
aDirectory|\newlogin.mdf;Integrated
Security=True;Connect Timeout=30"
        Dim con As New SqlConnection(conString)
        Dim hapus As String = "DELETE FROM
login WHERE username = @username"
        Dim cmd As New SqlCommand(hapus, con)

        cmd.Parameters.AddWithValue("@username",
        username.Text)
        Try
            con.Open()
            cmd.ExecuteNonQuery()
            LoginDataGridView.DataSource =

Nothing
            LoginDataGridView.Refresh()
            Dim str As String = "Select * FROM
login"

```

```

        cmd = New SqlCommand(str, con)
        Dim da As New SqlDataAdapter(cmd)
        Dim dt As New DataTable
        da.Fill(dt)
        LoginDataGridView.DataSource = dt
    Catch ex As Exception
        'MsgBox("Sorry, username sama",
MsgBoxStyle.OkOnly, "Invalid")
    Finally
        con.Close()
    End Try
    usrname.Focus()
End If
End Sub

Private Sub bupdate_Click(sender As Object, e
As EventArgs) Handles bupdate.Click
    If usrname.Text = "" Or psswd.Text = ""
Then
        MsgBox("Pilih username yang akan
diubah!")
    ElseIf MsgBox("Apakah anda yakin akan
mengubah password ini?") = MsgBoxResult.Ok Then
        Dim conString As String = "Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|Dat
aDirectory|\newlogin.mdf;Integrated
Security=True;Connect Timeout=30"
        Dim con As New SqlConnection(conString)
        Dim hapus As String = "UPDATE login SET
password = @psswd WHERE username = @usrname"
        Dim cmd As New SqlCommand(hapus, con)

        cmd.Parameters.AddWithValue("@psswd",
psswd.Text)
        cmd.Parameters.AddWithValue("@usrname",
usrname.Text)

        Try
            con.Open()
            cmd.ExecuteNonQuery()

```

```

        LoginDataGridView.DataSource =
Nothing
        LoginDataGridView.Refresh()
login"
        Dim str As String = "Select * FROM

        cmd = New SqlCommand(str, con)
        Dim da As New SqlDataAdapter(cmd)
        Dim dt As New DataTable
        da.Fill(dt)
        LoginDataGridView.DataSource = dt
    Catch ex As Exception
        'MsgBox("Sorry, username sama",
MsgBoxStyle.OkOnly, "Invalid")
    Finally
        con.Close()
    End Try
    usrname.Focus()
End If
End Sub
Private Sub
LoginDataGridView_CellClick_1(sender As Object, e
As DataGridViewCellEventArgs) Handles
LoginDataGridView.CellClick
    Dim baris As Integer
    With LoginDataGridView
        baris = .CurrentRow.Index
        usrname.Text = .Item(0, baris).Value
        psswd.Text = .Item(1, baris).Value
    End With
End Sub
End Class

```

D. *Listing Program Visual Basic Form Main Menu*

```

Imports System.IO.Ports
Imports System.Threading.Thread
Imports Microsoft.Office.Interop.Excel
Imports
System.Windows.Forms.DataVisualization.Charting

```

```

Public Class fMenu
    'Inisialisasi port
    Dim WithEvents COMPort As New SerialPort
    'Inisialisasi komunikasi data
    Private readBuffer As String = String.Empty
    Private ByteNumber As Integer
    Private ByteToRead As Integer
    Private byteEnd(2) As Char
    'Inisialisasi parsing data
    Dim strinput As String
    Dim data(4) As String
    Dim max As Double
    Dim putus As String
    'Inisialisasi waktu
    Dim Now As String
    'Inisialisasi save excel
    Dim d1() As String
    Dim d2() As String
    Dim d3() As String
    Dim timerCounter As Integer = 0

    Dim jmldatasama As Integer = 200 'data yang
    dibandingkan
    Dim jmldatactr As Integer = 0
    Dim range As Integer = 100 'range nilai yang
    dibandingkan
    Dim rangepts As Integer = 20 'range nilai yang
    dibandingkan
    Dim dtsebelum As Double = Double.MinValue
    Dim titikluluh As Double = Double.MaxValue

    Dim fresult As New formdata()
    Dim prosesoff As Boolean = False
    Dim disconnect As Boolean = False

    Private Sub fMenu_Load(sender As Object, e As
    EventArgs) Handles MyBase.Load
        Try
            ReDim d1(1)
            ReDim d2(1)

```

```

        ReDim d3(1)
        For Each COMString As String In
My.Computer.Ports.SerialPortNames
            cmbport.Items.Add(COMString)
        Next
        cmbport.SelectedIndex = 0
        cmbbaud.SelectedItem = "9600"
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
    chartfilter.Titles.Add("Data Serial Real
Time")
    Chart1.Titles.Add("Data Serial Real Time")
End Sub
Private Sub btnconnect_Click(sender As Object,
e As EventArgs) Handles btnconnect.Click
    If btnconnect.Text = "Connect" Then

        COMPort.BaudRate =
Val(cmbbaud.SelectedItem)
        COMPort.PortName = cmbport.SelectedItem
        Try
            COMPort.Open()
            Label8.Text = "Connected"
            btnconnect.Text = "Disconnect"
            GroupBox1.Enabled = True
            GroupBox2.Enabled = False
            GroupBox3.Enabled = True
            GroupBox4.Enabled = True
            GroupBox5.Enabled = True
            tb_clear()
        Catch ex As Exception

        End Try
    Else
        COMPort.Close()
        Label8.Text = "Disconnected"
        btnconnect.Text = "Connect"

        Timer1.Enabled = False
    End If
End Sub

```

```

        System.Threading.Thread.Sleep(10)
        disconnect = True

        GroupBox1.Enabled = False
        GroupBox2.Enabled = True
        GroupBox3.Enabled = False
        GroupBox4.Enabled = False
        GroupBox5.Enabled = False
        chartfilter.Series.Item("Moving Average
Filter").Points.Clear()
        Chart1.Series.Item("No
Filter").Points.Clear()
        tb_clear()
    End If

End Sub

Private Sub Receiver(ByVal sender As Object,
ByVal e As SerialDataReceivedEventArgs) Handles
COMPort.DataReceived

    If COMPort.IsOpen Then
        Try
            byteEnd =
COMPort.NewLine.ToCharArray
            Bytenumber = COMPort.BytesToRead
            readBuffer = COMPort.ReadLine()
            Me.Invoke(New
EventHandler(AddressOf DoUpdate))

            Catch ex As Exception
            End Try
        End If
    End Sub

    Public Sub DoUpdate(ByVal sender As Object,
ByVal e As System.EventArgs)

        Try
            Call proses_fix()
            'max

```

```

        Dim dt As Double = data(2)
        If dt > max Then
            max = dt
        End If
        dtsebelum = dt

    Catch ex As Exception
    Finally
    End Try

    If disconnect Then
        prosesoff = True
    End If
End Sub
Private Sub btnstart_Click(sender As Object, e
As EventArgs) Handles btnstart.Click
    If btnstart.Text = "Start" Then
        Try
            COMPort.RtsEnable = True
            COMPort.DtrEnable = True
            System.Threading.Thread.Sleep(10)

            btnstart.Text = "Stop"
            COMPort.Write("a")
            Timer1.Enabled = True
            btreset.Enabled = False
            max = Double.MinValue
        Catch ex As Exception

        End Try
    Else
        COMPort.RtsEnable = False
        COMPort.DtrEnable = False
        System.Threading.Thread.Sleep(10)

        btnstart.Text = "Start"
        COMPort.Write("b")
        btreset.Enabled = True

        Dim max_ As String

```



```

max_ = (max / 100).ToString("F2")

tbmax.Text = max_

Module1.dt1 = d1
Module1.dt2 = d2

Module1.pu = max_
Module1.delta = data(1)
Module1.engChart =
chartfilter.Series.Item("Moving Average Filter")
Module1.nofilter =
Chart1.Series.Item("No Filter")

Timer1.Enabled = False
tbdtawal.Text = " "

End If
End Sub
Private Sub proses_fix()
    strinput = tbdtawal.Text
    Dim panjang_data As Integer
    Dim x As Integer
    Dim z As Integer
    panjang_data = Len(strinput)

    Dim i As Integer
    i = 0
    z = 0

    For x = 1 To Len(readBuffer$)
        If Mid(readBuffer$, x, 1) = "|" Then
            z = z + 1
            data(i) = Mid(readBuffer$, z, x -
z)

            i = i + 1
            z = x
        End If
    Next x
End Sub

```

```

    Private Sub cmbPort_SelectedIndexChanged(sender
As Object, e As EventArgs) Handles
cmbport.SelectedIndexChanged
    If COMPort.IsOpen = False Then
        COMPort.PortName = cmbport.Text
    Else
        MsgBox("Valid only if port is Closed",
vbCritical)
    End If
End Sub

    Private Sub cmbBaud_SelectedIndexChanged(sender
As Object, e As EventArgs) Handles
cmbbaud.SelectedIndexChanged
    If COMPort.IsOpen = False Then
        COMPort.BaudRate = cmbbaud.Text
    Else
        MsgBox("Valid only if port is Closed",
vbCritical)
    End If
End Sub

    Private Sub Timer1_Tick(sender As Object, e As
EventArgs) Handles Timer1.Tick
    Try
        Now = TimeOfDay
        tbdtawal.Text = readBuffer
        tbdtx.Text = data(1)
        tbdtty.Text = data(2)
        tbcoba.Text = data(3)
        Label15.Text = Format(TimeOfDay,
"hh:mm:ss")
        chartfilter.Series.Item("Moving Average
Filter").Points.AddXY(data(1), data(3))
        Chart1.Series.Item("No
Filter").Points.AddXY(data(1), data(2))
    Catch ex As Exception
    End Try

    If tbdtawal.Text <> "" Then
        timerCounter = timerCounter + 1
    End If
End Sub

```

```

        d1(timerCounter - 1) = data(1)
        d2(timerCounter - 1) = data(3)
        d3(timerCounter - 1) =
Format(TimeOfDay, "hh:mm:ss")

        ReDim Preserve d1(timerCounter + 1)
        ReDim Preserve d2(timerCounter + 1)
        ReDim Preserve d3(timerCounter + 1)
    End If
    'End If
End Sub
Private Sub tb_clear()
    tbdawal.Text = " "
    tbdtx.Text = " "
    tbdty.Text = " "
    'tbluluh.Text = " "
    tbmax.Text = " "
    tbcoba.Text = " "
End Sub
Private Sub ClosePort()
    If COMPort.IsOpen Then
        COMPort.Close()
    End If
End Sub
Private Sub btnmethhod_Click(sender As Object,
e As EventArgs) Handles btnmethhod.Click
    finformation.Show()
End Sub

Private Sub btnhasil_Click(sender As Object, e
As EventArgs) Handles btnhasil.Click
    formdata.Show()
End Sub

Private Sub Button1_Click(sender As Object, e
As EventArgs) Handles Button1.Click
    Dim xlApp As Application = New
Application()

```

```

        If xlApp Is Nothing Then
            MsgBox.Show("Excel is not properly
installed!!")
            Return
        End If

        Dim sfd As New SaveFileDialog
        If sfd.ShowDialog() = DialogResult.OK Then
            Dim xlWorkBook As Workbook
            Dim xlWorkSheet As Worksheet
            Dim misValue As Object =
System.Reflection.Missing.Value

            xlWorkBook = xlApp.Workbooks.Add()
            xlWorkSheet = xlWorkBook.Sheets(1)

            Dim row As Long = d1.Length
            Dim arr(row + 1, 3) As Object

            arr(0, 0) = "WAKTU"
            arr(0, 1) = "DATA X"
            arr(0, 2) = "DATA Y"
            For i As Long = 1 To row
                arr(i, 0) = d3(i - 1)
                arr(i, 1) = d1(i - 1)
                arr(i, 2) = d2(i - 1)
            Next

            Dim sCell As Range =
xlWorkSheet.Cells(1, 1)
            Dim eCell As Range =
xlWorkSheet.Cells(row, 3)
            Dim rng As Range =
xlWorkSheet.Range(sCell, eCell)

            rng.Value2 = arr

            xlWorkBook.SaveAs(sfd.FileName)
            xlWorkBook.Close(True, misValue,
misValue)

```

```

xlApp.Quit()

chartfilter.Series.Clear()
End If

ReDim d1(1)
ReDim d2(1)
ReDim d3(1)
timerCounter = 0
End Sub
Private Sub Button2_Click(sender As Object, e
As EventArgs) Handles Button2.Click
    Dim mystream As New IO.MemoryStream
    chartfilter.Serializer.Save(mystream)
    Dim temp As New
DataVisualization.Charting.Chart
    temp.Serializer.Load(mystream)

    If temp.Titles.Count > 0 Then
        With temp
            .Titles(0).Font = New
System.Drawing.Font("Calibri", 18, FontStyle.Bold)
            .Series("Moving Average
Filter").BorderWidth = 5

        .Printing.PrintDocument.DefaultPageSettings.Landsca
pe = True

            .Printing.PrintPreview()
        End With
    End If
End Sub
Private Sub Button3_Click(sender As Object, e
As EventArgs) Handles Button3.Click
    Close()
End Sub
Private Sub btreset_Click(sender As Object, e
As EventArgs) Handles btreset.Click
    COMPort.Close()
    Timer1.Enabled = False
    tb_clear()

```

```

        chartfilter.Series.Item("Moving Average
Filter").Points.Clear()
        Chart1.Series.Item("No
Filter").Points.Clear()
        If readBuffer <> " " Then
            COMPort.Open()
            readBuffer = 0
        End If
    End Sub
End Class

```

E. *Listing Program Visual Basic Form Information*

```

Imports System.Data.SqlClient
Public Class finformation

    Public tes As New formdata()

    Dim _order As String
    Dim _project As String
    Dim _date1 As Date
    Dim _penguji As String
    Dim _sample As String
    Dim _dtlo As String
    Dim _dtd As String
    Dim _diameter As String
    Dim _panjang As String
    Dim _lebar As String

    Private Sub clearfb()
        tbd.Text = " "
        tbdiameter.Text = " "
        tbexaminer.Text = " "
        tblength.Text = " "
        tblo.Text = " "
        tborder.Text = " "
        tbproject.Text = " "
        tbsample.Text = " "
        tbwidth.Text = " "
    End Sub

```

```

        End Sub
        Private Sub Button1_Click(sender As Object, e
As EventArgs) Handles Button1.Click

            If tborder.Text = "" Or tbproject.Text = ""
Then
                MessageBox.Show("Isi yang bertanda *",
"Information", MessageBoxButtons.OK)
            Else
                MessageBox.Show("Data berhasil
disimpan!", "Information", MessageBoxButtons.OK)

                Module1.order = tborder.Text
                Module1.project = tbproject.Text
                Module1.date1 = DateTimePicker1.Text
                Module1.penguji = tbexaminer.Text
                Module1.sample = tbsample.Text
                Module1.dtlo = tblo.Text
                Module1.dtd = tbd.Text
                Module1.diameter = tbdiameter.Text
                Module1.panjang = tblength.Text
                Module1.lebar = tbwidth.Text

            End If
            cleartb()
        End Sub

        Private Sub Button2_Click(sender As Object, e
As EventArgs) Handles Button2.Click
            Me.Close()
        End Sub
    End Class

```

F. *Listing Program Visual Basic Form Result*

```

Imports System.Data.SqlClient
Imports Microsoft.Office.Interop.Excel

```

```

Public Class formdata

    'Inisialisasi hasil rumus
    Dim tleleh As Double
    Dim tultimat As Double
    Dim tdelta As String
    Dim tluas As String
    Dim telongasi As Double

    Private Sub formdata_Load(sender As Object, e
As EventArgs) Handles MyBase.Load

        lborder.Text = Module1.order
        lbproject.Text = Module1.project
        tbhdate.Text = Module1.date1
        tbhpenguji.Text = Module1.penguji
        tbhkode.Text = Module1.sample
        tbhlo.Text = Module1.dtlo
        tbhd.Text = Module1.dtd
        tbhpu.Text = Module1.pu
        tbhdelta.Text = delta
    End Sub

    Private Sub Button3_Click(sender As Object, e
As EventArgs) Handles btsave.Click
        Dim xlApp As Application = New
Application()

        If xlApp Is Nothing Then
            MessageBox.Show("Excel is not properly
installed!!")
            Return
        End If

        Dim sfd As New SaveFileDialog
        If sfd.ShowDialog() = DialogResult.OK Then
            Dim xlWorkBook As Workbook
            Dim xlWorkSheet As Worksheet
            Dim misValue As Object =
System.Reflection.Missing.Value

```



```

xlWorkBook = xlApp.Workbooks.Add()
xlWorkSheet = xlWorkBook.Sheets(1)

Dim size = 100
Dim arr(size, size) As Object

arr(0, 0) = "Order"
arr(0, 1) = "Project"
arr(0, 2) = "Titik MAX"
arr(1, 0) = lborder.Text
arr(1, 1) = lbproject.Text
arr(1, 2) = tbhpu.Text

Dim sCell As Range =
xlWorkSheet.Cells(1, 1)
Dim eCell As Range =
xlWorkSheet.Cells(size, size)
Dim rng As Range =
xlWorkSheet.Range(sCell, eCell)

rng.Value2 = arr

xlWorkBook.SaveAs(sfd.FileName)
xlWorkBook.Close(True, misValue,
misValue)
xlApp.Quit()
End If
End Sub

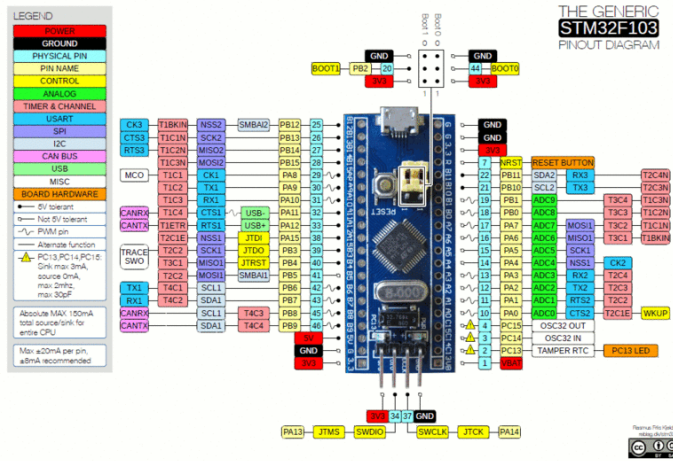
Private Sub btback_Click(sender As Object, e As
EventArgs) Handles btback.Click
Close()
End Sub

End Class

```

-----Halaman ini sengaja dikosongkan-----

A. Datasheet Mikrokontroller STM32F103C8T6





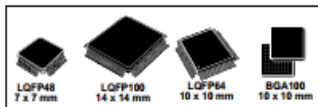
STM32F103x6 STM32F103x8 STM32F103xB

Performance line, ARM-based 32-bit MCU with Flash, USB, CAN, seven 16-bit timers, two ADCs and nine communication interfaces

Preliminary Data

Features

- Core: ARM 32-bit Cortex™-M3 CPU
 - 72 MHz, 90 DMIPS with 1.25 DMIPS/MHz
 - Single-cycle multiplication and hardware division
 - Nested interrupt controller with 43 maskable interrupt channels
 - Interrupt processing (down to 6 CPU cycles) with tail chaining
- Memories
 - 32-to-128 Kbytes of Flash memory
 - 6-to-20 Kbytes of SRAM
- Clock, reset and supply management
 - 2.0 to 3.6 V application supply and I/Os
 - POR, PDR, and programmable voltage detector (PVD)
 - 4-to-16 MHz quartz oscillator
 - Internal 8 MHz factory-trimmed RC
 - Internal 32 kHz RC
 - PLL for CPU clock
 - Dedicated 32 kHz oscillator for RTC with calibration
- Low power
 - Sleep, Stop and Standby modes
 - V_{BAT} supply for RTC and backup registers
- 2 x 12-bit, 1 µs A/D converters (16-channel)
 - Conversion range: 0 to 3.6 V
 - Dual-sample and hold capability
 - Synchronizable with advanced control timer
 - Temperature sensor
- DMA
 - 7-channel DMA controller
 - Peripherals supported: timers, ADC, SPIs, I²Cs and USARTs



- Debug mode
 - Serial wire debug (SWD) & JTAG interfaces
- Up to 80 fast I/O ports
 - 32/49/80 5 V-tolerant I/Os
 - All mappable on 16 external interrupt vectors
 - Atomic read/modify/write operations
- Up to 7 timers
 - Up to three 16-bit timers, each with up to 4 IC/OC/PWM or pulse counter
 - 16-bit, 6-channel advanced control timer:
 - up to 6 channels for PWM output
 - Dead time generation and emergency stop
 - 2 x 16-bit watchdog timers (Independent and Window)
 - SysTick timer: a 24-bit downcounter
- Up to 9 communication interfaces
 - Up to 2 x I²C interfaces (SMBus/PMBus)
 - Up to 3 USARTs (ISO 7816 interface, LIN, IrDA capability, modem control)
 - Up to 2 SPIs (18 Mbit/s)
 - CAN interface (2.0B Active)
 - USB 2.0 full speed interface

Table 1. Device summary

Reference	Root part number
STM32F103x6	STM32F103C6, STM32F103R6
STM32F103x8	STM32F103C8, STM32F103R8
STM32F103xB	STM32F103V8
STM32F103x6	STM32F103R6 STM32F103V6

RIWAYAT PENULIS



Nama : Juniar Adi Prasetya
TTL : Jombang, 18 Juni 1996
Jenis Kelamin : Laki - Laki
Agama : Islam
Alamat Asal : Perum. Sambong Permai
D-05, Jombang, Jawa Timur

Telp/HP : 081330623620
E-mail : juniaradii@gmail.com

RIWAYAT PENDIDIKAN

- 2002 – 2008 : SDN Jombang 2
- 2008 – 2011 : SMPN 1 Tembelang
- 2011 – 2014 : SMAN Ploso
- 2015 – Sekarang : Departemen Teknik Elektro
Otomasi Institut Teknologi
Sepuluh Nopember.

PENGALAMAN KERJA

- Kerja Praktek di PT Indonesia Power UP, Denpasar, Bali

PENGALAMAN ORGANISASI

- Panitia IARC (Industrial Automation and Robotic Competition) Publikasi 2016
- Panitia IARC (Industrial Automation and Robotic Competition) Publikasi 2017
- Staff Departemen HMKI Mahasiswa Himpunan Mahasiswa D3 Teknik Elektro ITS 2015 – 2016
- Kabiro Departemen HMKI Mahasiswa Himpunan Mahasiswa D3 Teknik Elektro ITS 2016 – 2017

-----Halaman ini sengaja dikosongkan-----